COMPUTATIONAL FLUID DYNAMICS FOR NAVAL ENGINEERING PROBLEMS

THÈSE N^{O} 3138 (2004)

PRÉSENTÉE À LA FACULTÉ SCIENCES DE BASE

Institut d'analyse et de calcul scientifique

SECTION DE MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Nicola PAROLINI

Laurea in Ingegneria Aerospaziale, Politecnico di Milano, Italie
et de nationalité italienne

acceptée sur proposition du jury:

Prof. A. Quarteroni, directeur de thèse
Prof. M. Falcone, rapporteur
Prof. J. Rappaz, rapporteur
M. G. Simmer, rapporteur

Lausanne, EPFL
2004

Abstract

The subject of this thesis is the numerical simulation of viscous free-surface flows in naval engineering applications. State-of-the-art numerical methods based on the solution of the Navier–Stokes equations are used to predict the flow around different classes of boats.

We investigate the role of the Computational Fluid Dynamics in the design of racing boats, such as America's Cup yachts and Olympic class rowing hull. The mathematical models describing the different aspects of the physical problem, as well as the numerical methods adopted for their solution, are introduced and critically discussed. The different phases of the overall numerical simulation procedure, from grid generation through the solution of the flow equations to the post-processing of the results, are described.

We present the numerical simulations that have been performed to investigate the role of different design parameters in the conception of America's Cup yachts and we describe how the results obtained from the simulations are integrated into the overall design process.

The free-surface flow around an Olympic rowing boat is also considered. We propose a simplified approach to take into account the effect of the boat dynamics in the prediction of the hydrodynamic forces acting on the boat. Based on the results of the simulations, we propose a new design concept and we investigate its potential benefits on the boat performances.

One of the aspects that is found to be not completely satisfactory, within the standard numerical methods adopted, is the modelling of complex free-surface flows. The second part of this thesis is devoted to a more theoretical and methodological investigation of this aspect.

In particular, we present and analyse a new numerical method based on the level set approach for the solution of two-fluid flows. The numerical scheme based on a finite element discretization is introduced and different critical aspects of its implementation are discussed. In particular, we present and analyse a new technique for the stabilization of the advection equation associated to the level set problem. Moreover, we propose a new reinitialization procedure for the level set function which plays a crucial role in the accuracy of the algorithm. The convergence properties of this procedure are analysed and comparisons with more standard approaches are presented.

Finally, the proposed method has been used to solve a variety of test cases concerning time dependent two-fluid viscous flows. The results of the simulation are presented and discussed.

Riassunto

Questa tesi si occupa della simulazione numerica di flussi viscosi a superficie libera per applicazioni in ingegneria navale. Metodi numerici avanzati per la soluzione delle equazioni di Navier–Stokes vengono utilizzati per predire il flusso attorno a diverse tipologie di scafi.

Viene proposta un'analisi del ruolo della simulazione numerica nel processo di sviluppo di imbarcazioni da competizione, quali barche a vela di classe America e scafi da canottaggio di classe Olimpica. I modelli matematici e i metodi numerici adottati per la descrizione e la simulazione del problema fisico vengono introdotti e viene proposta un'analisi critica delle loro potenzialità e dei loro limiti.

Si presentano i risultati ottenuti dalle simulazioni realizzate nell'ambito di uno studio riguardante l'analisi di diversi parametri di progetto nella messa a punto di una barca a vela di classe America. Si descrive inoltre le modalità di integrazione dei risultati delle simulazioni numeriche nel ciclo di progetto.

In seguito, viene considerato il flusso a superficie libera attorno ad uno scafo da canottaggio di classe Olimpica, per il quale viene proposto un metodo semplificato per studiare l'impatto della dinamica dell'imbarcazione nella stima delle forze idrodinamiche agenti sull scafo. I risultati delle simulazioni hanno portato alla definizione di un'innovazione progettuale di cui vengono investigati i possibili benefici sulle prestazioni dello scafo.

Dalle simulazioni realizzate è emerso che uno degli aspetti che possono ritenersi non completamente soddisfacenti, nell'approccio numerico adottato, è la modellizzazione di flussi a superficie libera. Nella seconda parte di questo lavoro, si propone un'indagine più teorica e metodologica di questo aspetto.

Viene presentato un nuovo schema numerico basato sull'approccio a curve di livello (*level set*) per la soluzione di flussi bifasici basato su una discretizzazione agli elementi finiti. Alcuni aspetti critici legati alla sua implementazione vengono analizzati. In particolare, si propone un nuovo metodo per la stabilizzazione dell'equazione di trasporto associata alla soluzione del problema *level set*. Le proprietà di convergenza e di conservazione della massa del metodo proposto vengono analizzate e confrontate con quelle di metodi di stabilizzazione classici. Nell'approccio *level set* un ruolo importante è giocato dalla cosiddetta procedura di *reinizializzazione*. A tale proposito, viene proposta una nuova tecnica per la quale viene condotta un'analisi di convergenza.

Infine, il metodo introdotto è stato utilizzato per la simulazione di diversi flussi bifasici e i risultati delle simulazioni vengono presentati e discussi.

A Lucia

Acknowledgments

The first person who deserves my gratitude is Prof. Alfio Quarteroni for giving me the opportunity to work on a very interesting subject and in a stimulating research environment. His precious advises and the confidence he has always demonstrated in me have been essential for my scientific and personal growth during these years.

I thank the members of the jury, Prof. Maurizio Falcone, Prof. Jacques Rappaz and Grant Simmer, for having carefully read this manuscript and for their helpful suggestions to improve it. Prof. Charles A. Stuart is also acknowledged for having presided the jury.

A special thank to Erik Burman for the long afternoons spent at the black board discussing about most of the theoretical analyses presented in this work. His constant and patient attitude in introducing me into the theory of finite elements has been fundamental for the progress of my research.

With Geoffrey W. Cowles and Mark L. Sawley, and more recently with Davide Detomi, I have shared the exciting experience of working on the optimization of an America's Cup racing yacht, in the framework of the collaboration between the EPFL and the Alinghi Team. I learned much from them about the way to face the intrinsic difficulties of such a challenging problem and to respect unpostponable deadlines. The members of the Alinghi Design Team, in particular Jim Bungener, Michael Richelsen and Manolo Ruiz de Elvira, are also acknowledged for their precious support and their patience in explaining to us the various aspects of the racing yacht design procedure. A special thank to Prof. Jan-Anders Månson and Pascal Vuilliomenet for their ability in managing the EPFL/Alinghi project and for the helpful support that I have constantly received from them.

I thank all the researchers, assistants and students who during these years have been members of the Chair of Modelling and Scientific Computing (CMCS), as well as all the members of the Institute of Analysis and Scientific Computing (IACS), for the many fruitful discussions on different scientific (and non-scientific) topics. Special thanks to my three favorite system administrators, Simone Deparis, Diego Mastalli and Klaus Sapelza, who patiently supported my Linux activity and to Mme Jacqueline Mosetti for her invaluable assistance. This work has benefited from the collaboration with Andrea Mola, Benedetto Conserva, Stefano Pandini and Olivier Grandjean, who carried out their diploma and semester projects under my supervision. I thank them for their precious contributions.

After a few years, I still feel in debt with Prof. Luigi Quartapelle and Franco Auteri, my advisor and co-advisor during the Master thesis. They introduced me into the "magic" world of numerical simulations for fluid dynamics. I thank them for the constant support and friendship.

I should thank here many friends (in Switzerland, Italy and elsewhere) who prevented me from falling into a nerdy life as a Ph.D. student in mathematics. The list would be too long and for sure incomplete. Anyway, they are on the list of people to whom I will offer to drink.

A special thanks to my family, Mamma Mariarosa, Babbo Giulio, Nonna Lidia, Laura, Marco, Chiara and Francesco, for being the wonderful family they are.

Finally, thank you, Lucia, for your love. Sappi che la cosa è reciproca.

Contents

Introduction **1**

1 Mathematical Models for Naval Engineering Problems **9**

1.1 Flow equations 9

1.1.1 Initial and boundary conditions 12

1.2 Toward the numerical approximation: free-surface models 13

1.2.1 Front Tracking methods 14

1.2.2 Front Capturing methods 15

1.3 Modelling the turbulence 17

1.3.1 Turbulence models 19

1.3.2 Laminar-turbulent transition 21

1.3.3 Wall treatment 21

2 Numerical Discretization based on Finite Volumes **23**

2.1 Integral form of the flow equations 23

2.1.1 Finite volume discretization 24

2.2 Discretization of the flow equations 25

2.2.1 SIMPLE algorithm 26

2.2.2 Time discretization 27

2.3 Overview of the solution algorithm 28

2.4 Parallel approach 29

3 Numerical Simulations for Yacht Design **31**

3.1 Introduction 31

3.2 An overview on America's Cup yacht design 31

3.3 The mathematical problem and its complexity 35

3.3.1 Devising hierarchical models to reduce complexity 36

3.3.2 Further simplified models 36
3.4 Grid generation 37
3.5 Numerical simulations of the flow around yacht appendages 40
3.5.1 Bulbs 41
3.5.2 Winglets 47
3.6 Numerical simulations of the flow around yacht sails 57
3.7 Scalability results of parallel computations 60

4 Numerical Simulation of Free Surface Viscous Flows 63
4.1 A three-dimensional free-surface test case 64
4.2 Wigley hull test case 69

5 Numerical Simulation on Olympic Class Rowing Boats 73
5.1 Introduction 73
5.2 Numerical simulations on steady configurations 75
5.2.1 Comparison with experimental results 75
5.2.2 Different hull designs 78
5.3 A reduced model for rowing boat dynamics 79
5.4 Numerical simulations with imposed pitching motion 81
5.4.1 Moving grid approach 81
5.4.2 Numerical results 83
5.5 Introduction of pitch stabilizers 85
5.5.1 Design of the pitch stabilizer 85
5.5.2 Forces on the isolated stabilizer 87
5.5.3 Effect of the pitch stabilizer on the global performances 89

6 A Level Set Finite Element Method for the Solution of Free-Surface Flows 93
6.1 Problem setting 94
6.2 Finite Element discretization 95
6.2.1 Discretization of the surface tension term 97
6.3 Time discretization and inexact factorization schemes 100
6.3.1 Incremental Chorin–Temam projection scheme 102
6.3.2 Incremental Yosida projection scheme 102
6.4 Overview of the solution algorithm 103

7 Subgrid Edge Stabilization for the Level Set equation 105

7.1 A steady advection/reaction model problem 106

7.2 The subgrid viscosity stabilization 107

7.3 An edge-based subgrid viscosity stabilization 109

7.3.1 Two-level $\mathbb{P}_1$ interpolation 109

7.3.2 Two-level $\mathbb{P}_2$ interpolation 112

7.4 Numerical examples 114

7.4.1 Advection-reaction test case 115

7.4.2 Linear advection problem 115

7.4.3 Solid body rotation 119

8 The Level Set Reinitialization 121

8.1 Standard reinitialization procedures 122

8.1.1 Direct reinitialization 122

8.1.2 PDE-based reinitialization 122

8.2 Interface local projection reinitialization 123

8.2.1 Interface local projection reinitialization algorithm 123

8.2.2 Abstract framework 125

8.2.3 Convergence analysis of the interface local projection reinitialization . . 131

8.3 Numerical test cases 135

8.3.1 Mass conservation test 135

8.3.2 Effect of reinitialization on two-fluid flow problems 137

9 Numerical Results of Two-Fluid Flow Simulations 141

9.1 Bubble deformation under surface tension action 141

9.2 Rising Bubble 144

9.3 Rayleigh-Taylor instability 149

9.4 Drop falling into a free-surface 153

9.5 Broken dam problem 154

Conclusions 157

Introduction

The objective of Computational Fluid Dynamics (CFD) in naval engineering applications is to accurately simulate the behaviour of full scale ships in real operating conditions. In the past few decades, the development of accurate and effective numerical methods for the simulation of the flow around an advancing boat has been subject of intensive research activities in both the academic and industrial communities. As long as the simulation capabilities improve, the need of expensive and time-demanding experimental tests will reduce more and more. This trend is already present in the naval industry, as well as in many other fields such as, *e.g.*, aeronautical and automotive industries.

The role of Computation Fluid Dynamics acquires a particular relevance in those applications where optimal design is critical. This is the case, *e.g.*, for racing boats such as International America's Cup Class (IACC) yachts. It is well-known in the America's Cup community that a constant evolution and improvement in design is required to lead the competition. As a matter of fact, the same yacht that dominates one edition of the America's Cup has no hope to win, just three or four years later, the next edition, unless substantial design improvements are achieved. This implies that the design process is subjected to severe time constraints and, in this respect, there is an increasing demand of accurate and effective numerical tools that could (at least partially) replace the expensive towing tank and wind tunnel measurements. Indeed, many design choices made by an America's Cup team are still based on real full-scale tests, where two boats with different configurations sail one against the other and the relative performances are measured. This approach allows the experienced sailors to "feel" the improvement (or the deterioration) of the performances associated with a given design change. However, factors such as the environmental noise (differences in wind and sea-state between the two boats) and the human element of how the sails are trimmed and the boat steered, can represent serious drawbacks. Besides full scale tests, each design team performs experimental tests (in water tank and wind tunnel) and simulations based on different numerical methods with various levels of accuracy, complexity, and computational cost.

The earliest important theoretical results concerning ship hydrodynamics date back to the end of nineteenth century, when the studies carried on by Froude and Kriloff gave rise to a first general theory about ship motion. They built the bases for the understanding of the different relevant physical aspects in the dynamics of an advancing vessel. In particular, Froude investigated the role of the wave component of the resistance and gave his name to the non-dimensional number that characterizes free-surface flows (the Froude number). In 1898, a thin ship theory was proposed in [Mic98] to describe the wave generated by an advancing ship and an analytical estimation of the wave resistance (based on the so-called *Mitchell's integral*) was derived.

More recently, numerical methods based on the solution of the potential flow equation were developed in the field of aerodynamics (see, *e.g.*, the pioneering work by Hess and Smith [HS67])

and later extended to the solution of ship hydrodynamic problems [Daw77a, Daw77b]. Although they are based on a simple irrotational and inviscid flow model, potential flow panel codes are still commonly used in the ship hydrodynamics community [RLDS93].

In the last two decades, in parallel with the increase of available computational power, numerical methods based on the solution of the Navier–Stokes (and Euler) equations have been successfully applied to naval engineering problem [Hin92, FMJ93]. The solution of Navier–Stokes equations has broadened the class of problems that could be faced, including the possibility of treating viscous, turbulent and separated flows.

The first objective of this thesis is to investigate the role of Computational Fluid Dynamics for naval engineering problems. In particular, we want to show how the design of a vessel can benefit from the adequate use of numerical techniques currently available. We consider two applications in highly competitive frameworks, such as the America's Cup sailing competition and Olympic class rowing races. In both cases, optimal design is critical and small details can make the difference between a winning and a losing boat. We will present several numerical investigations on the influence of design parameters which are commonly considered critical for the optimization of these kind of boats. Moreover, we will show how these results can be used in the actual definition of the boat configuration.

A critical analysis of these results obtained using state-of-the-art CFD techniques allowed us to identify some aspects of the numerical algorithms that still deserve investigations.

One such aspect, the accurate tracking of the free-surface in two-fluid flows, represents the second objective of this thesis. A more methodological and theoretical work has been carried out in order to develop a new method for the solution of two-fluid flows. Although the current implementation of this method is limited to the solution of two-dimensional flows, we believe that the techniques that we propose here represent an advance in the treatment of free-surface problems. The extension of the proposed method to three dimensional problems, which is currently in progress, will be applied in the next future to the solution of ship hydrodynamic problems.

This Ph.D. dissertation consists of two parts. In the first part, we make use of advanced numerical methods for the solution of the Navier–Stokes equations to simulate the flow around different classes of racing boats, analysing the influence of several design parameters and showing how numerical simulations can be integrated into the overall design process. In the second part, we focus our attention on the free-surface aspect and we introduce and analyse a new numerical scheme for the solution of two-fluids flows.

In **Chapter 1**, we derive the mathematical model which describes the free-surface flow around an advancing boat. The model is based on the density-dependent (inhomogeneous) Navier–Stokes equations, which read

$$\partial_t \rho + \boldsymbol{u} \cdot \boldsymbol{\nabla} \rho = 0, \tag{1}$$

$$\rho \partial_t \boldsymbol{u} + \rho(\boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u})\boldsymbol{u} - \boldsymbol{\nabla} \cdot \boldsymbol{T}(\boldsymbol{u}, p) = \rho \boldsymbol{g} + \boldsymbol{f}_\Gamma, \tag{2}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \tag{3}$$

where $\boldsymbol{u}$ and p are the velocity and pressure fields, $\boldsymbol{T}(\boldsymbol{u},p) = \mu(\boldsymbol{\nabla u} + \boldsymbol{\nabla u}^T) - p\boldsymbol{I}$ is the stress tensor, ρ and μ are variable density and viscosity coefficients and $\boldsymbol{f}_\Gamma$ is a source term associated to the surface tension.

An overview of possible approaches for the solution of free-surface flows is given, with special emphasis on the family of *front-capturing* methods. These methods are based on an Eulerian approach for the solution of the flow equations. We consider, in particular, the Volume of Fluid (VOF) method [HN81, PZ99, MP99, PP04, Azc02a] and the Level Set method [OS88, ZCMO96, Set99, VL99, Smo01, OF02, IC03].

Both methods are based on a suitable splitting of problem (1)-(3). The momentum equation (2) and the incompressibility constraint (3) are decoupled from the mass conservation equation (1) and the latter is replaced by an advection equation for either a discontinuous function (the volume fraction, for the Volume of Fluid method) or a continuous function (the signed distance from the interface, for the Level Set method). In this work, both methods have been used and the advantages and disadvantages associated to each of them are discussed through the whole dissertation.

Typically, the Reynolds numbers that characterize ship hydrodynamics problems are large (order of 10^6–10^7). The flow is turbulent around a large portion of the boat and suitable turbulence models have to be used to estimate correctly the forces acting on it. A Direct Numerical Simulation (DNS) of such flows is unaffordable with the computational power available today. In Chapter 1, we give an overview of the different turbulence models that can be used in the framework of a Reynolds Averaged Navier–Stokes solver, with particular emphasis on the model ($k-\epsilon$) that will be used in the simulations.

A numerical discretization of the mathematical model based on a finite volume spatial discretization is presented in **Chapter 2**. The schemes adopted for the computation of fluxes and for the solution of the Navier–Stokes system, as well as the algorithms used for the solution of the algebraic problems, are rather standard and are briefly recalled. The discretization of the transport equation for the volume fraction and of the equations associated to the turbulence model are also described.

In **Chapter 3**, the numerical scheme introduced is used for the numerical simulation of the flow around an America's Cup yacht. An extensive simulation campaign has been carried out in the framework of a collaboration between our Institute and the Swiss Team Alinghi in preparation for the 2003 edition of the America's Cup. During this activity, a simulation procedure for the analysis of several aspects of the design has been set up. The different simulation phases (geometry reconstruction, grid generation, solution of the flow equation and post-processing of the results) are described. In particular, we discuss the grid generation approach that we have adopted, based on hybrid grids, in order to guarantee an accurate solution of the problem as well as the robustness required when dealing with complex geometries.

A full understanding of the hydrodynamic flow around an America's Cup yacht is far from being achieved. The complex interactions between the different components of the boats, under and over the water surface, as well as the strict constraints imposed by the America's Cup rules, make the design of a performing boat an extremely difficult task. In the last few years, numerical results on America's Cup yacht design have been given in [Mil98, THD01, JK01, DRRF02].

In this work, we try to assess the potential impact that the use of advanced numerical methods can have in the overall design process. We consider the design of the yacht appendages and we investigate some of the most critical parameters defining their geometry. In this framework, the dependence of the different parameters on the boat performances is estimated by comparing different configurations. In particular, we have analysed different bulb and winglet designs and we

have simulated their performances in different sailing conditions. The definition of performance indices for the ranking of different configuration is also discussed. For this study, the effect of the free surface has been neglected.

However, the free-surface dynamics can not be neglected when one focuses on the hydrodynamic analysis of the hull. Indeed, the component of the resistance on the hull associated to the wave generation can represent a dominant percentage of the global resistance.

The influence of an accurate prediction of the free-surface dynamics on performance estimations is discussed in **Chapter 4**. Comparisons with experimental results on three-dimensional test cases are presented in order to validate the reliability and accuracy of the method here adopted.

The same numerical approach has been used to simulate the free-surface flow around an Olympic Class rowing boat. Steady state simulations on different rowing hulls and time-depending simulations, with the hull subjected to an imposed motion, are presented. In particular, we focus our attention on the analysis of the additional drag contribution due to the pitching motion of the boat. We propose a simplified approach for the coupling between the viscous flow solution and a reduced model which accounts for the complex dynamics that the rowing boat undergoes during the race. We also discuss possible design improvements resulting from the simulation analysis.

Although the Volume of Fluid method that was adopted in the numerical simulation was found to be reasonably accurate in predicting hull performances, the numerical results showed that this method suffers from a lack of accuracy on the free-surface reconstruction. As a matter of fact, a smearing of the interface on several cells is observed. This problem is due to the fact that the Volume of Fluid method adopted makes use of low order schemes in the solution of the advection equation for the discontinuous volume fraction field. To overcome these difficulties, one possible approach makes use of interface reconstruction techniques based on purely geometrical consideration. Examples are the donor-acceptor algorithm [RT76, FVB99], the SLIC algorithm [NW76, MPR03] and the PLIC algorithm [AP91, Rud97, PP04]. However, the implementation of these techniques for three dimensional problems, in particular when unstructured grids are employed, is not always straightforward and the computational cost for large three-dimensional computations can become excessively high.

An alternative approach for the solution of free-surface flows is proposed in **Chapter 6**. This approach is based on the Level Set method, where the interface is defined as the zero level set of a continuous function ϕ, usually identified with the signed distance function from the interface. In the past few years, this method has been extensively used for the numerical solution of interface problems in different domains, from free-surface and mean curvature flows to image processing. Most of the numerical schemes proposed in the literature are based on finite difference spatial discretizations and are restricted to cartesian structured grids.

There exist just a few examples of Level Set implementations using finite element discretization [Tor00, Smo01, GRR04]. One of the objective of this work is to show how this combination can lead to the definition of an accurate and effective tool for the solution of free-surface problems involving complex interface topologies on computational domains with general shape.

In our formulation, the continuity equation (1) is replaced by an advection equation for the level set function ϕ:

$$\phi_t + \boldsymbol{u} \cdot \boldsymbol{\nabla}\phi = 0, \tag{4}$$

and the density and viscosity coefficients in (2) are computed based on the sign of ϕ.

System (2)-(3)-(4) is discretized in space using ($\mathbb{P}_1$-iso$\mathbb{P}_2$)-$\mathbb{P}_1$ finite elements for the Navier–Stokes equations and $\mathbb{P}_1$ elements for the level set equation. The latter is decoupled from the Navier–

Stokes system and a semi-implicit time discretization is adopted. This leads to the solution, at each time step, of a Navier–Stokes problem with a given interface position (the one computed at the previous time step) followed by the solution of the advection level set equation based on the velocity field just computed. The Navier–Stokes system is solved using an algebraic projection method based on inexact factorization schemes [QSV99, QSV00].

Numerical difficulties are encountered when solving a pure advection equation, such as (4), with standard Galerkin-based finite-element methods [Joh87]. Spurious oscillations can appear which affect the convergence properties of the scheme. The remedy consists either in resorting to discontinuous Galerkin methods [LR74], for which stability and local mass conservation is built into the method or using a stabilized Galerkin formulation for the continuous finite elements. Several different stabilization techniques have been proposed in the literature in the past thirty years. For an overview on classical stabilization techniques, we refer to [Mor96, QV94].

In a recent paper [Bur04], it was shown that it is possible to construct interior penalty methods which are stable and of optimal order for advection/reaction/diffusion equations for approximation spaces of continuous functions. One possibility is to add to the standard Galerkin approximation a term penalizing the jump of the gradient over the internal edges of the triangulation.

In **Chapter 7**, we introduce a new stabilization technique which inherits the idea of internal penalty from [Bur04] and recasts it into the subgrid framework proposed in [Gue99a]. A consistent term penalizing the jump of the gradient across the internal edges of element patches is added to the standard Galerkin formulation. We prove that this term enjoys the same stability and convergence properties as the subgrid viscosity considered in [Gue99a] and quasi-optimal error estimates for the stabilized problem hold. Moreover, this approach yields a slightly more economic scheme than the one proposed in [Gue99a].

This approach has the advantage of being independent of the time discretization, when compared to more classical stabilization techniques, such as, *e.g.*, SUPG [BH82], and high-order time-stepping schemes can easily be adopted. In level set problems, the solution of the advection equation by time-stepping schemes with order higher than one can have a great impact, in particular with respect to the mass conservation properties of the method. Numerical results highlighting this aspect are presented.

The level set function is usually initialized as the signed distance function from the initial position of the interface. However, during advection, this function undergoes large deformations that can lead to a flattening (and/or steepening) of the function. These deformations, in particular flat regions close to the interface, have a negative impact on the accuracy of the scheme. In the level set literature, it has been pointed out by several authors [DT80, SSO94, CS02, CHMO96] that a suitable redistancing procedure has to be adopted in order to avoid this undesired behaviour.

This procedure is often referred to as *reinitialization* and consists in periodically replacing the level set function with the signed distance function from the current interface position. Different techniques have been proposed to accomplish this step, some of them [Smo01] are based on a direct computation of the distance of each grid node from the interface, others [SSO94, Tor00] are based on the solution of an additional Hamilton–Jacobi type problem.

In **Chapter 8**, we review the role of the reinitialization step in the level set method and we propose a new technique based on a special treatment of the region close to the interface, where optimal accuracy is required in order to avoid an artificial displacement of the interface. We show how to construct the best (in the L^2-sense) continuous piecewise linear approximation of the signed distance function in the interface region and we obtain a convergence result for the proposed

method. Numerical tests and comparisons with standard reinitialization procedures are presented and discussed.

The need of a combined use of a stabilization for the level set advection equation and a reinitialization of the level set function can be argued. Indeed, the solution of pure advection equations with standard (non stabilized) Galerkin approximation is critical in presence of regions with large gradient of the solution, where spurious oscillations can be generated. However, the reinitialization procedure gives an intrinsic control on the gradient since for a distance function ϕ, we have $\|\boldsymbol{\nabla}\phi\| = 1$. On the other hand, the steepening of flat section of the interface has an effect of "anti"-diffusion and may lead to numerical instabilities. This dichotomy was not pointed out in previous works which adopted both stabilization of the level set equation and reinitialization. In fact, stabilization and reinitialization have two different roles: the former has to guarantee the accuracy of the scheme even in presence of large gradients in the solution, the latter has to avoid the flattening of the function in proximity of the interface.

The level set method proposed in this work has been implemented in a finite element library. In its current state, laminar free-surface flows in two-dimensions can be simulated. In **Chapter 9**, we present a set of numerical simulations carried out using this method. Different test cases have been designed in order to assess the mathematical properties (stability, accuracy, robustness) of the method when dealing with a wide range of free-surface problems. In particular, the algorithm for the computation of the surface tension terms are tested by solving curvature driven interface problems. Different test cases concerning time dependent two-fluid flows, *i.e.* rising bubbles, falling drops, Rayleigh instability and broken dam flows, are presented and discussed.

The structure of this dissertation is somehow unusual: a first part reporting extensive numerical results is followed by a second part where a numerical scheme is proposed and theoretical analyses are carried out. Indeed, the main motivation underlying the development of the method introduced in the second part stems from a critical analysis of the results obtained in the first part, which highlighted the limitations of commonly adopted numerical schemes when dealing with complex free-surface problems. Although the numerical results presented in Chapter 9 are restricted to laminar free-surface flows in two dimensions, the methodology here proposed can be extended to three-dimensional problems.

To help the reader following the logical flow of this work, in Fig. 1 we present an outline of the thesis structure illustrating the interrelations between the different chapters.

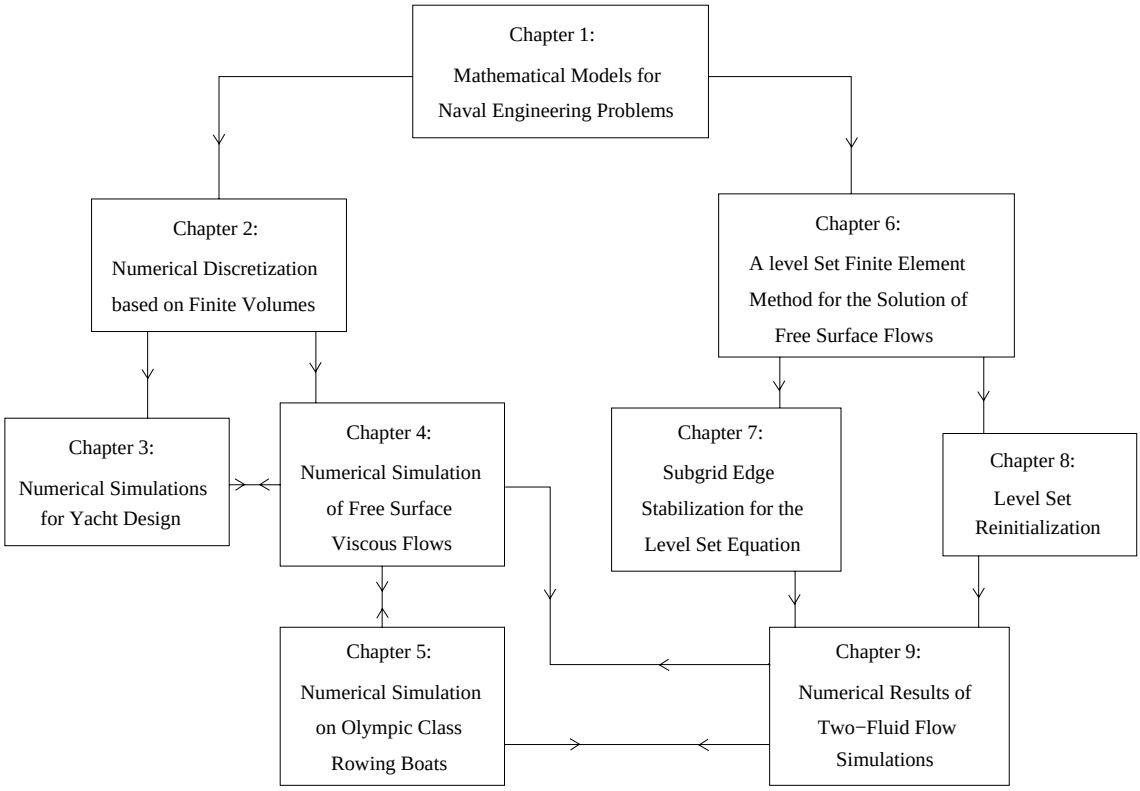


Figure 1: Overview of the thesis plan.

Chapter 1

Mathematical Models for Naval Engineering Problems

Ideally, a global complete mathematical model for ship hydrodynamics should be able to reproduce the many aspects of the physics underlying the motion of a boat advancing in the water. In order to simulate accurately the hydrodynamic and aerodynamic flow around the boat, the model should account for the viscous effects, the transitional (laminar to turbulent) nature of the flow as well as the wave generation on the water free-surface. Sometimes, the interaction between the fluid and the structure (water-hull and air-sails in the case of a sailing boat) should be considered as well.

In the present work, we will consider the turbulent viscous free-surface flows around advancing boats, neglecting the fluid-structure interaction aspect. In this chapter a suitable mathematical model for this problem is derived. Moreover, modelling aspects related to the presence of a free-surface and to the turbulent nature of the flow are introduced and discussed.

The analysis carried out in this chapter was already addressed in part in [PQ04], where the mathematical and numerical models adopted for the simulation of the flow around an America's cup yacht, including the fluid-structure interaction between the sails and the air, were discussed.

1.1 Flow equations

We start recalling the physical assumptions used to derive a mathematical description for the considered problem. We consider two fluids (water and air), both assumed to be viscous, newtonian and immiscible, surrounding a boat in steady advancing motion in the water. The air and water flows are incompressible and isothermal, so that the viscosity and density changes due to temperature variations can be neglected. These assumption are fully justified for the considered problem, since the flow around boats in standard navigation conditions is always characterized by low Mach number and negligible temperature gradients. We also suppose that the two fluids are separated by a sharp interface (with density and viscosity having a jump discontinuity at the interface). The surface tension effects are taken into account in the model, even if, in most naval engineering applications, they can be neglected.

We consider a bounded domain $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ surrounding the boat, with the artificial external boundaries far enough from the boat in order to avoid problems related to wave reflections. At each

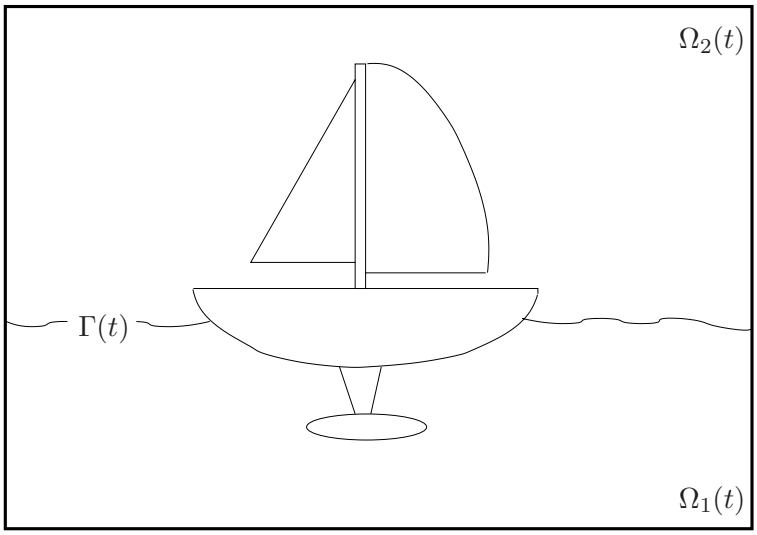


Figure 1.1: A two-dimensional section of the computational domain $\Omega = \text{int}(\overline{\Omega_1(t) \cup \Omega_2(t)})$

time $t > 0$, two different fluids, fluid 1 (*i.e.*, water) and fluid 2 (*i.e.*, air), fill respectively the sub-domains $\Omega_1(t)$ and $\Omega_2(t)$ such that $\Omega = \text{int}(\overline{\Omega_1(t) \cup \Omega_2(t)})$ (see Fig. 1.1 for a two-dimensional sketch).

In each of the two sub-domains $\Omega_i(t), i = 1, 2$, the flow is governed by the incompressible Navier–Stokes equations:

$$\rho_i \partial_t \boldsymbol{u}_i + \rho_i (\boldsymbol{u}_i \cdot \boldsymbol{\nabla}) \boldsymbol{u}_i - \boldsymbol{\nabla} \cdot \boldsymbol{T}_i(\boldsymbol{u}_i, p_i) = \rho_i \boldsymbol{g}, \tag{1.1}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u}_i = 0, \tag{1.2}$$

where $\boldsymbol{u}_i$ is the velocity field, p_i is the pressure, ρ_i is the density, $\boldsymbol{g} = (0, 0, g)^T$ is the gravity acceleration, and $\boldsymbol{T}_i(\boldsymbol{u}_i, p_i) = \mu_i(\boldsymbol{\nabla}\boldsymbol{u}_i + \boldsymbol{\nabla}\boldsymbol{u}_i{}^T) - p_i \boldsymbol{I}$ is the stress tensor with μ_i indicating the molecular viscosity. Subscript i indicates that all the quantities are restricted to the sub-domain Ω_i and time derivatives are denoted by ∂_t. Equation (1.1) enforces the conservation of linear momentum, while equation (1.2) is the constraint of incompressibility which enforce the mass conservation in each sub-domain Ω_i.

The free-surface Γ is a sharp interface between Ω_1 and Ω_2. Since there is no flow through it, the normal components of the two velocities $\boldsymbol{u}_1 \cdot \boldsymbol{n}$ and $\boldsymbol{u}_2 \cdot \boldsymbol{n}$ should agree on Γ. Furthermore, the tangential components must match as well since the two flows are viscous. Thus we have the following kinematic condition

$$\boldsymbol{u}_1 = \boldsymbol{u}_2, \qquad \text{on } \Gamma. \tag{1.3}$$

Moreover, the forces acting on the fluid at the free-surface are in equilibrium. This is a dynamic condition and means that the normal forces on either side of Γ are of equal magnitude and opposed direction, while the tangential forces must agree in both magnitude and direction:

$$\boldsymbol{T}_1(\boldsymbol{u}_1, p_1) \cdot \boldsymbol{n} = \boldsymbol{T}_2(\boldsymbol{u}_2, p_2) \cdot \boldsymbol{n} + \kappa\sigma\boldsymbol{n} \quad \text{on } \Gamma, \tag{1.4}$$

where σ is the surface tension coefficient, which is a property of the liquid and depends on the temperature. Here, we consider it constant. The quantity κ in (1.4) is twice the mean curvature of

the free-surface, such that

$$\kappa = R_{t_1}^{-1} + R_{t_2}^{-1},$$

where R_{t_1} and R_{t_2} are radii of curvature along the coordinates (t_1, t_2) of the plane tangential to the free-surface (orthogonal to $\boldsymbol{n}$).

Problem (1.1)-(1.2) can be rewritten as a single set of density-dependent (or inhomogeneous) Navier-Stokes equations holding on the entire domain Ω:

$$\partial_t \rho(\boldsymbol{x}) + \boldsymbol{u} \cdot \boldsymbol{\nabla} \rho(\boldsymbol{x}) = 0, \tag{1.5}$$

$$\rho(\boldsymbol{x}) \partial_t \boldsymbol{u} + \rho(\boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}) \boldsymbol{u} - \boldsymbol{\nabla} \cdot \boldsymbol{T}(\boldsymbol{u}, p) = \rho \boldsymbol{g} + \boldsymbol{f}_\Gamma, \tag{1.6}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0, \tag{1.7}$$

where $\boldsymbol{u}$, p and $\boldsymbol{T}(\boldsymbol{u}, p) = \mu(\boldsymbol{x})(\boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla} \boldsymbol{u}^T) - p\boldsymbol{I}$ are now defined in the whole Ω and $\rho(\boldsymbol{x})$ and $\mu(\boldsymbol{x})$ are variable density and viscosity coefficients. With respect to problem (1.1)-(1.2), the additional equation (1.5) is introduced to express the mass conservation over the whole domain Ω, with the incompressibility constraint (1.7) remaining valid. Equations (1.5)-(1.7) have to be interpreted in the sense of distributions, given the discontinuous nature of $\rho(\boldsymbol{x})$ and $\mu(\boldsymbol{x})$.

Note that in (1.6) an additional source term $\boldsymbol{f}_\Gamma$ appears. It accounts for the jump on the normal stress tensor in the dynamic interface condition (1.4) and is defined by:

$$\boldsymbol{f}_\Gamma = \kappa \sigma \delta_\Gamma \boldsymbol{n}, \tag{1.8}$$

where δ_Γ is the Dirac delta function with support on Γ. For a formal derivation of equation (1.8), we refer to [BKZ92].

System (1.5)-(1.7) can be rewritten in conservation form, as follows:

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = 0, \tag{1.9}$$

$$\partial_t (\rho \boldsymbol{u}) + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) - \boldsymbol{\nabla} \cdot \boldsymbol{T}(\boldsymbol{u}, p) = \rho \boldsymbol{g} + \boldsymbol{f}_\Gamma, \tag{1.10}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0. \tag{1.11}$$

The above equations have to be complemented with suitable initial and boundary conditions (see 1.1.1).

A global existence result can be proven for the solution of (1.5)-(1.7) provided Ω is a bounded, connected, open subset of $\mathbb{R}^3$ with smooth boundary (the latter condition is not satisfied in the case at hand, indeed the boundary is only Lipschitz-continuous). In that case, if $\rho_0 \in L^\infty(\Omega)$ and $\boldsymbol{u}_0 \in (H^1(\Omega))^3$, $\boldsymbol{m}_0 = \rho \boldsymbol{u}|_{t=0} \in L^\infty(\Omega)$ then a weak solution exists (see [Lio97]) which satisfies

$$\begin{aligned} \rho &\in L^\infty(0, T; L^\infty(\Omega)), \\ \boldsymbol{u} &\in L^2(0, T; H_0^1(\Omega))^3, \\ \rho |\boldsymbol{u}|^2 &\in L^\infty(0, T; L^1(\Omega)), \\ \boldsymbol{\nabla} \boldsymbol{u} &\in L^2(\Omega \times (0, T)), \\ \rho &\in C([0, T]; L^p(\Omega)), \quad \forall \quad 1 \leq p < \infty. \end{aligned} \tag{1.12}$$

Moreover, the following energy inequalities hold

$$\begin{aligned} \frac{d}{dt}\int_\Omega \rho|\boldsymbol{u}|^2\,d\boldsymbol{x} + \int_\Omega \mu(\partial_i u_j + \partial_j u_i)^2\,d\boldsymbol{x} \leq \\ 2\int_\Omega \rho\boldsymbol{g}\cdot\boldsymbol{u}\,d\boldsymbol{x} \quad \text{in } \mathcal{D}'(0,T), \end{aligned} \tag{1.13}$$

$$\begin{aligned} \int_\Omega \rho|\boldsymbol{u}|^2\,d\boldsymbol{x} + \int_0^t\int_\Omega \mu(\partial_i u_j + \partial_j u_i)^2\,d\boldsymbol{x}\,ds \leq \\ \int_\Omega |\boldsymbol{m}_0|^2/\rho_0\,d\boldsymbol{x} + 2\int_0^t\int_\Omega \rho\boldsymbol{g}\cdot\boldsymbol{n}\,d\boldsymbol{x}\,ds \\ \text{a.e. } t\in(0,T), \end{aligned} \tag{1.14}$$

where ∂_i denotes partial derivative w. r. to x_i, $\mathcal{D}'(0,T)$ is the space of distributions on $(0,T)$ and the summation convention on repeated indexes applies.

Uniqueness of weak solutions is known to be an open problem; however, any weak solution is equal to a strong solution if the latter exists. Uniqueness is closely related to the regularity of solutions. In this respect, there is no further meaningful regularity result on $\boldsymbol{u}$ and ρ other than what stated in (1.12), in particular very little is known on the pressure field p (which acts as a Lagrange multiplier in (1.6)-(1.7)).

1.1.1 Initial and boundary conditions

Suitable initial and boundary conditions have to be imposed to close problem (1.5)-(1.7). The initial conditions for velocity and density are given as follows

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x},0) &= \boldsymbol{u}_0, \quad \forall \boldsymbol{x}\in\Omega, \\ \rho(\boldsymbol{x},0) &= \rho_0, \quad \forall \boldsymbol{x}\in\Omega, \end{aligned}$$

where $\boldsymbol{u}_0$ is a divergence-free velocity field.

Regarding boundary conditions for the velocity, we consider a subdivision of the boundary $\Sigma = \partial\Omega$ in four regions:

- an *inflow* region Σ_{in}, where a Dirichlet boundary condition is imposed

$$\boldsymbol{u}|_{\Sigma_{\text{in}}} = \boldsymbol{u}_{\text{in}}(t), \quad \forall t\in]0,T];$$

- an *outflow* region Σ_{out}, where a zero normal stress boundary condition is imposed:

$$\boldsymbol{T}(\boldsymbol{u},p)|\cdot\boldsymbol{n}|_{\Sigma_{\text{out}}} = \boldsymbol{0}, \quad \forall t\in]0,T],$$

  where $\boldsymbol{n}$ isi the unit outward normal on Σ_{out};

- a *wall* region Σ_{w}, where a no-slip Dirichlet boundary condition is imposed:

$$\boldsymbol{u}|_{\Sigma_{\text{w}}} = \boldsymbol{u}_w(t), \quad \forall t\in]0,T];$$

- a *symmetry* region Σ_{sym}, where a symmetry boundary condition is imposed:

$$\begin{aligned} \boldsymbol{u}|\cdot\boldsymbol{n}|_{\Sigma_{\text{sym}}} &= 0, \quad \forall t \in]0,T], \\ \boldsymbol{\nabla}(\boldsymbol{u}-(\boldsymbol{u}\cdot\boldsymbol{n})\boldsymbol{n})|\cdot\boldsymbol{n}|_{\Sigma_{\text{sym}}} &= 0, \quad \forall t \in]0,T], \end{aligned}$$

  which states that the normal velocity component is zero as well as the normal derivatives of the tangential velocity.

Note that, if Dirichlet boundary conditions on the velocity are imposed on the whole boundary $\Sigma_{\text{in}} = \Sigma$, the initial and boundary conditions should satisfy additional compatibility conditions (see [Qua93]).

A boundary condition on ρ has to be prescribed only at the inflow Σ_{in}:

$$\rho(\boldsymbol{x},t)|_{\Sigma_{\text{in}}} = \rho_{in}(t), \qquad \forall t \in]0,T],$$

where we have assumed that no back-flow is present on the outflow region, such that

$$\Sigma_{\text{in}} = \{\boldsymbol{x} \in \Sigma \mid (\boldsymbol{u}\cdot\boldsymbol{n}) < 0\}.$$

1.2 Toward the numerical approximation: free-surface models

The accurate prediction of the free-surface around an advancing boat is a crucial aspect of ship hydrodynamics. The initial position of the water-air interface is known but its evolution in time has to be computed as part of the solution of problem (1.5)-(1.7).

Several numerical methods for the solution of free-surface problems have been proposed in the literature in the past few decades. These methods can be classified based on their ability in treating different physical situations (*i.e.* flow regimes or types of waves) and on their computational complexity. The choice of the most suitable approach depends therefore on the specific problem at hand.

A popular (although not exhaustive) classification of the numerical methods for free-surface problems divides them in two main categories:

- *Front Tracking methods*: the free-surface interface is explicitly tracked along the trajectory of the fluid particles, making use of the *kinematic* interface condition (1.3). These methods are usually based on a Lagrangian of mixed Eulerian-Lagrangian approach, where the computational grid is adapted to the interface and must be readjusted each time the free-surface is moved (see Fig. 1.2, left).

- *Front Capturing methods*: the free-surface interface is reconstructed from the properties of an appropriate field function (*e.g.* phase volume fraction or density). These methods are based on an Eulerian approach: the computational grid is fixed and both the regions occupied by liquid and gas are modelled (see Fig. 1.2, right).

We refer to [Smo01] for a discussion on the different approaches and for a classification of methods based on the different aspects of the numerical models (flow model, interface model, flow-interface coupling, discretization methods).

1.2.1 Front Tracking methods

Usually, in *Front Tracking methods*, only the liquid phase is computed and the free-surface is treated as a boundary of the computational domain defined as height function (see, *e.g*, [NH73, Hym84, TBE$^+$01]):

$$z = h(x, y, t),$$

which is the distance between the free-surface and a reference line (in 2D) or surface (in 3D). This approach can be adopted as long as h is a single-valued function. This means that complex interface problems, such as bubble merging or wave breaking, cannot be handled.

The free-surface motion is governed by the kinematic condition (1.3), as follows

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial x} + v\frac{\partial h}{\partial y} = w, \tag{1.15}$$

where u, v and w are the cartesian components of the velocity field. The dynamic free-surface condition (1.4) is then imposed as boundary condition on h and system (1.1)-(1.2) is solved in $\Omega_1(t)$.

In this case, the standard approach for the solution of the bulk flow is based on recasting the flow equations (1.1)-(1.2), together with free-surface conditions (1.3)-(1.4), into a curvilinear coordinate system. The grid points are then moved to fit the free-surface shape [MP97]. The mesh motion can represent another critical aspect since, when large free surface deformations occur, the grid elements can become highly skewed, which is usually a problem for the stability and accuracy of the Navier–Stokes solvers. In these cases, partial or global remeshing becomes necessary.

In the past years, front tracking type algorithms have been the dominant methods in naval engineering applications. Some examples of successful use of these methods in marine hydrodynamics can be found in [Hin92, FMJ93, BJWT99]. However, their intrinsic limitations when dealing with complex free-surface topologies opened the field to the development of alternative approaches.

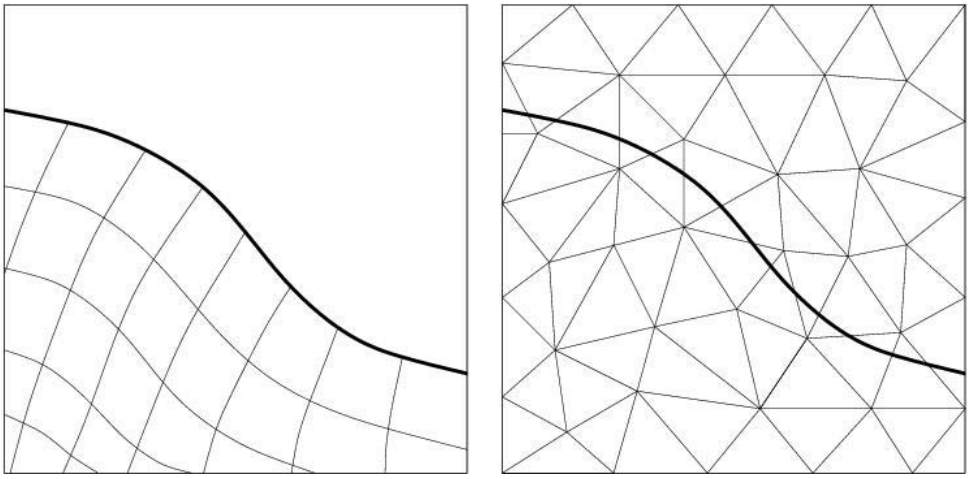


Figure 1.2: Typical grid topologies in 2D for Lagrangian (left) and Eulerian (right) free-surface methods. The thick line represents the free-surface.

1.2.2 Front Capturing methods

Due to their flexibility in dealing with complex free-surface problems, *Front Capturing methods* are receiving an increasing attention in the naval engineering community.

A first approach of this type was considered in the so-called *Marker-and-Cell* method introduced in [HW65]. In this method, particles with neither mass nor energy are distributed in the whole fluid domain to track the free-surface location. These particles do not play any role in the dynamics of the fluid and are not taken into account in the solution of the flow equations.

Indeed, the Marker-and-Cell method was the first approach able to handle complex and general situations (such as breaking surfaces, splash and fluid detachment) and its original implementation has later been improved by several authors (see, *e.g.*, [Miy86, TM94, NM96]). On the other hand, its computational cost for the solution of large three-dimensional problems is prohibitive, because of the need of a large number of particles to capture the free-surface shape properly. Moreover, to avoid the generation of false regions of void, the particles need to be periodically redistributed.

The most commonly employed Front Capturing methods for the simulation of free-surface flows with complex interfaces are the so-called *Volume of Fluid* (VOF) method and *Level Set* method. For both, the computational grid is fixed and both the regions occupied by liquid and gas are usually modelled. The interface between the different immiscible fluids is "captured" by solving an additional advection equation for a suitable field variable.

Indeed, both methods are based on a splitting of problem (1.5)-(1.7). The momentum equation (1.6) and the incompressibility constraint (1.7) are decoupled from the mass conservation equation (1.5). The latter is replaced by an advection equation for either a discontinuous function (the volume fraction, for the Volume of Fluid method) or a continuum function (the signed distance from the interface, for the Level Set method). In the VOF method the interface is identified as a *discontinuity line* of the volume fraction, while in the Level Set method the interface is implicitly represented by the zero Level Set of the signed distance function.

Volume of Fluid method

In the VOF method, originally introduced in [HN81], the dynamics of the interface is computed by advecting a function $\psi(\boldsymbol{x}, t)$ which represents the volume fraction of one phase (*e.g.* water) in each cell. The value of ψ is 1 in the cells completely filled by water and 0 in those filled by air. The cells where $0 < \psi < 1$ identify the *interface region* that should be kept as sharp as possible. This method requires the solution of a pure advection equation for the discontinuous function $\psi(\boldsymbol{x}, t)$, which reads

$$\frac{\partial \psi}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \psi = 0. \tag{1.16}$$

The Navier–Stokes equations can be either solved in the entire computational domain [Rud97] or only in the region occupied by water [MPR99]. If the first approach is used, the local values of density and viscosity are computed from the volume fraction, as follows

$$\begin{aligned} \rho(\boldsymbol{x}, t) &= \psi(\boldsymbol{x}, t)\, \rho_1 + (1 - \psi(\boldsymbol{x}, t))\, \rho_2, \\ \mu(\boldsymbol{x}, t) &= \psi(\boldsymbol{x}, t)\, \mu_1 + (1 - \psi(\boldsymbol{x}, t))\, \mu_2. \end{aligned} \tag{1.17}$$

Being ψ discontinuous, the numerical solution of equation (1.16) requires special care. If low order schemes are used, the numerical diffusion will lead to a smearing of the interface over several grid

elements. On the other hand, standard high-order schemes can lead to the appearance of oscillations in the interface region. To overcome these difficulties, one can resort to high resolution schemes for hyperbolic problems developed for the numerical solution of conservation laws [LeV92] (see, *e.g.*, the high-resolution interface capturing (HRIC) scheme proposed in [MP99] and the immersed interface method described in [LL97]). An alternative approach, extensively used in the Volume of Fluid literature, makes use of interface reconstruction techniques based on purely geometrical consideration. Examples are the donor-acceptor algorithm [RT76, FVB99], the SLIC algorithm [NW76, MPR03] and the PLIC algorithm [AP91, Rud97, PP04]. However, the implementation of these techniques for three dimensional problems, in particular when unstructured grids are employed, is not always straightforward and the computational cost for large three-dimensional computations can become excessively high. Another aspect that can be critical in VOF methods is the evaluation of the interface curvature, essential in applications where surface tension effects are relevant. Suitable algorithms for the reconstruction of the local curvature from the discontinuous volume fraction field have to be considered (see, *e.g.*, [Cab03]).

In the last decade, Volume of Fluid methods have been extensively used for marine hydrodynamic applications. They were first employed for internal flow problems such as tank sloshing [CA02] and green water flows [Nie03], and more recently to external flows around ship hulls (see, in particular, the works by Azcueta [Azc01, Azc02a, Azc02b]).

In the present work, this approach has been adopted for the solution of the viscous free surface flow around an Olympic class rowing boat, as will be discussed in Chapter 5. Moreover, validation numerical tests for the VOF method will be described in 4.

Level Set method

The Level Set method is based on the solution of the following advection equation:

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \phi = 0, \tag{1.18}$$

where ϕ is a smooth function defined in the whole computational domain as the signed distance function from the interface:

$$\begin{aligned} \phi(\boldsymbol{x}) &:= \text{dist}(\boldsymbol{x}, \Gamma), \qquad \forall \boldsymbol{x} \in \Omega_1, \\ \phi(\boldsymbol{x}) &:= -\text{dist}(\boldsymbol{x}, \Gamma), \qquad \forall \boldsymbol{x} \in \Omega_2. \end{aligned}$$

Negative values of ϕ correspond to fluid 1, while positive values to fluid 2. The zero level set of ϕ implicitly represents the interface.

The idea underlying this type of method was first proposed in [DT80], where the interface was defined as the zero level set of a continuous *pseudo-density* function. The Level Set method was introduced in [OS88] for the numerical solution of front propagating problems with curvature-dependent motion and then extended to a variety of physical applications. We refer to the two books [Set99] and [OF02] and to references therein for an overview of the numerical schemes and the description of a large collection of problems treated by the Level Set method.

The property of ϕ being a distance function is not preserved during advection. It has been shown in [SSO94] that, a *reinitialization* procedure is necessary in order to restore this property to the level set function, at least in regions close to the interface. As a result, this procedure enhances the performance of the numerical algorithm. This aspect will be discussed in details in Chapter 8.

In most of the Level Set literature, the finite difference spatial discretization is utilized (see, *e.g.*, [OS88, SSO94, SS97, SF99, CHMO96, ZCMO96]. Only more recently, finite element approximations have been considered by a few authors [Tor00, Smo01, GRR04].

The main advantage of the Level Set method, when compared with the VOF method, is that the advection equation (1.18) is solved for a continuous function, rather that a discontinuous one. Moreover, the evaluation of geometrical quantities, such as interface normals and curvature, is much easier. On the other hand, the VOF method guarantees better mass conservation properties [TLW95]. Indeed, in the Level Set method, the mass conservation properties strongly depend on the numerical schemes adopted for the solution of equation (1.18) and on the reinitialization procedure. In Chapter 6, we will introduce a Level Set method based on a finite element discretization. A suitable method for the solution of equation (1.18) and a new reinitialization technique are proposed and analysed in Chapters 7 and 8, respectively.

The Level Set method is receiving an increasing attention in the naval engineering community. Some examples of the use of this method for ship hydrodynamic simulations can be found in [VL99, BSHS99, CHS99, SD00, IDMC01, IC03]. In particular, in [VL99], the authors develop a one-phase formulation of the Level Set method, where only the water flow is computed. On the interface, *ad-hoc* interpolation schemes are used in order to prescribe the dynamic boundary condition. However, this formulation was found to be more difficult to implement (in particular in three dimensions) and less accurate than the standard two-phase formulation.

1.3 Modelling the turbulence

The flow around a boat in standard navigation conditions exhibits turbulent behaviour over the vast majority of the boat surface . Turbulent flows are characterized by being highly unsteady, three-dimensional, and containing vortices and coherent structures which stretch and increase the intensity of turbulence. Moreover, they fluctuate on a broad range of scales (in space and time). These features make the so-called direct numerical simulation (DNS) unaffordable. Indeed, in DNS the numerical solution of Navier–Stokes equations is based on space and time discretization fine enough to allow for all the significant structures of the turbulence to be captured. This would require the computational domain to be at least as large as the largest turbulent eddy, the latter being of the order of few times the linear length of the boat L. On the other hand, a direct simulation should be able to capture the kinetic energy dissipation, which occurs on the smallest scale whose size η has been determined by Kolmogoroff.

It turns out that in a DNS the number of grid points in each direction must be at least L/η, but this ratio is proportional to $Re_L^{3/4}$, where Re_L is the Reynolds number based on the magnitude of the velocity fluctuations. In the class of problem we are considering, the Reynolds number Re_L is of the order of 10^6. In the three-dimensional computational domain the total number of grid points should therefore scale as $Re_L^{9/4}$, a number which becomes easily prohibitive due to the limitations on computer speed and memory.

Reynolds averaged Navier–Stokes (RANS) models move from the idea of decomposing the velocity components u_i, the pressure p and the density ρ into a mean part and a fluctuating part

$$p = \bar{p} + p', \quad \rho = \bar{\rho} + \rho', \quad u_i = \bar{u_i} + u_i', \quad i = 1, 2, 3,$$

where the overbar denotes the Reynolds average, also called filter. The latter can be operated in a variety of ways, such as the Fourier filter (which is the truncation at a suitable order of the Fourier

series of a function), statistical average, time average for a statistically steady turbulence, volume average for a statistically homogeneous turbulence, or ensemble (space-time) average in a more general case.

Operating the average on the Navier–Stokes equations (1.5)-(1.7) and assuming that the filter satisfies the following properties

(i) *linearity*

$$\overline{u + \lambda v} = \bar{u} + \lambda \bar{v}, \qquad \forall \lambda \in \mathbb{C},$$

(ii) *commutation of derivatives and averages*

$$\overline{\partial_x u} = \partial_x \bar{u},$$

(iii) *double average* (filters have no effect on filtered variables)

$$\bar{\bar{u}} = \bar{u},$$

(iv) *product average*

$$\overline{u\bar{v}} = \bar{u}\bar{v},$$

we obtain:

$$\partial_t \bar{\rho} + \widetilde{\boldsymbol{u}} \cdot \boldsymbol{\nabla} \bar{\rho} = 0, \tag{1.19}$$

$$\partial_t(\bar{\rho}\widetilde{\boldsymbol{u}}) + \boldsymbol{\nabla}\cdot(\bar{\rho}\widetilde{\boldsymbol{u}} \otimes \widetilde{\boldsymbol{u}}) - \boldsymbol{\nabla}\cdot\boldsymbol{T}(\bar{\boldsymbol{u}}, \bar{p}) - \boldsymbol{\nabla}\cdot(\boldsymbol{R} + \boldsymbol{S}) = \rho \boldsymbol{f}, \tag{1.20}$$

$$\boldsymbol{\nabla}\cdot\bar{\boldsymbol{u}} = 0. \tag{1.21}$$

Here $\widetilde{\boldsymbol{u}} = \bar{\rho}\,\bar{\boldsymbol{u}}/\rho$ is the so-called Favre (density weighted) average, while $\boldsymbol{R}$ represents the Reynolds stress tensor

$$\boldsymbol{R}_{ij} = -\bar{\rho}\widetilde{u_i'' u_j''} = -\overline{\rho u_i'' u_j''}, \quad i,j = 1,2,3, \tag{1.22}$$

with the Favre fluctuating part defined as $\boldsymbol{u}'' = \boldsymbol{u} - \widetilde{\boldsymbol{u}}$; finally, the extra stress $\boldsymbol{S}_{ij} = \overline{\mu' \partial_j \boldsymbol{u}_i'}$, $i,j = 1,2,3$, is due to the fact that μ is not constant (for this derivation see e.g. [MP94]).

A complete derivation for the general case is long and tedious. A simpler form can be obtained when both ρ and μ are constant. This is actually the case when, for example, a domain decomposition approach is adopted, restricting the equations to the two sub-domains Ω_1 and Ω_2 (wherein density and viscosity are constant) and introducing suitable interface conditions (see Section 1.2). The formulation for the problem with constant coefficients (in either Ω_1 or Ω_2) is reported here for reader's convenience:

$$\partial_t(\rho\bar{\boldsymbol{u}}) + \boldsymbol{\nabla}\cdot(\rho\bar{\boldsymbol{u}} \otimes \bar{\boldsymbol{u}}) - \boldsymbol{\nabla}\cdot\boldsymbol{T}(\bar{\boldsymbol{u}}, \bar{p}) - \boldsymbol{\nabla}\cdot\boldsymbol{R} = \rho \boldsymbol{f}, \tag{1.23}$$

$$\boldsymbol{\nabla}\cdot\bar{\boldsymbol{u}} = 0. \tag{1.24}$$

where the Reynolds stress tensor reduces to $\boldsymbol{R}_{ij} = \rho\overline{u_i' u_j'}, i = 1,2,3$.

The new system of equations must be closed by introducing a suitable representation (or model) of the Reynolds stresses. Quoting D. C. Wilcox [Wil93]:

> *Herein lies the fundamental problem of turbulence. In order to compute all mean-flow properties of the turbulent flow under consideration, we need a prescription for computing* $\overline{u_i' u_j'}$.

1.3.1 Turbulence models

Several different models for the Reynolds stresses have been proposed in the literature, having different complexities and efficiencies. They can be roughly divided into three main categories:

- algebraic models;
- models based on differential equations (Spalart–Allmaras, $k-\epsilon$, $k-\omega$);
- models based on transport equations for the Reynolds Stress Tensor components;

The methods belonging to the first two categories are based on the so-called *Boussinesq hypothesis* (see [Bou97]), assuming that the turbulent stress tensor can be expressed as a linear function of the strain rate tensor, in analogy with Stokes law for laminar flows, as follows:

$$\boldsymbol{R} = \mu_T \left(\boldsymbol{\nabla}\bar{\boldsymbol{u}} + \boldsymbol{\nabla}\bar{\boldsymbol{u}}^T \right) - \frac{2}{3}\rho\, k\, \boldsymbol{I}, \tag{1.25}$$

where μ_T is referred to as the *turbulent eddy viscosity* and $k = \frac{1}{2}|\overline{\boldsymbol{u}'}|^2$ is the turbulent kinetic energy.

Algebraic models

In *algebraic* models, μ_T is computed from the mean-flow variables and the geometric length scale, by means of algebraic relations derived from experimental data. The most relevant of these models is the *mixing length model* proposed by Prandtl, where the eddy viscosity is supposed to be proportional to the product between the mixing length l_m and a turbulent velocity scale depending on the mean flow $v_T = v_T(\bar{\boldsymbol{u}})$, namely

$$\mu_T \approx \rho\, l_m\, v_T. \tag{1.26}$$

For three dimensional flows, the most successful algebraic model is the one proposed by Baldwin and Lomax, see [BL78]. Supplying with their simplicity to their weak physical foundation, these models can give good results in problems characterized by simple geometry and by strong experimental background, such as jets or boundary layers developing on flat plates.

Another strategy for the closure of problem (1.23)-(1.24) consists in expressing the eddy viscosity in terms of one or more turbulent quantities, each calculated by solving a transport equation, to be added to the original problem.

Spalart–Allmaras model

In the *Spalart–Allmaras model* (see [SA92]), one additional transport equation for the eddy viscosity μ_T is considered. The equation for μ_T reads:

$$\partial_t \mu_T + \overline{\boldsymbol{u}} \cdot \boldsymbol{\nabla}\mu_T = G_\mu + \frac{1}{\sigma_\nu}\left[\boldsymbol{\nabla}\cdot\left((\mu+\mu_T)\boldsymbol{\nabla}\mu_T\right)\right) + C_2\left(\boldsymbol{\nabla}{\mu_T}^2\right)\Big] - Y_\mu, \tag{1.27}$$

where σ_μ and C_2 are constants, while G_μ and Y_μ are terms accounting for, respectively, the production and dissipation of turbulent viscosity and have to be suitably modelled. These quantities are expressed in terms of the mean flow variables and based on empirical or statistical considerations.

k-ϵ model

The k-ϵ *model* [LS72] is a two-equation model, in which two transport equations are added to the original problem, one for the turbulent kinetic energy k and one for its dissipation rate ε which is defined as

$$\epsilon = \frac{\mu}{\rho}\overline{\boldsymbol{\nabla u}' \cdot (\boldsymbol{\nabla u}' + \boldsymbol{\nabla u}'^T)}.$$

The two additional equations for k and ϵ are obtained by averaging moments of the Navier–Stokes equation and by successively modelling the unknown terms resulting by this operation. These equations can be written as follows

$$\rho\partial_t k + \rho\bar{\boldsymbol{u}} \cdot \boldsymbol{\nabla} k = \boldsymbol{\nabla} \cdot [(\mu + \mu_T/\sigma_k)\boldsymbol{\nabla} k] + G_k + G_b - \rho\epsilon, \tag{1.28}$$

$$\rho\partial_t \epsilon + \rho\bar{\boldsymbol{u}} \cdot \boldsymbol{\nabla} \epsilon = \boldsymbol{\nabla} \cdot [(\mu + \mu_T/\sigma_\epsilon)\boldsymbol{\nabla} \epsilon] + C_{\epsilon 1}\frac{\epsilon}{k}(G_k + C_{\epsilon 3}G_b) - C_{\epsilon 2}\frac{\rho\epsilon^2}{k}, \tag{1.29}$$

where G_k accounts for the production of turbulent kinetic energy due to the mean velocity gradients, G_b is the production of turbulent kinetic energy due to buoyancy, σ_k and σ_ϵ are constants.

After solving equations (1.28) and (1.29), the eddy viscosity can be computed as

$$\mu_T = C_\mu \frac{\rho\epsilon^2}{k}. \tag{1.30}$$

Typical values for the coefficients appearing in equations (1.28), (1.29) and(1.30) are $C_{\epsilon 1} = 1.44, C_{\epsilon 2} = 1.92, C_\mu = 0.09, \sigma_k = 1.0$ and $\sigma_\epsilon = 1.3$.

Different versions of the k-ϵ model have been proposed in the literature, each characterized by specific expressions for the production and dissipation terms. In particular, the so-called *realizable* k-ϵ is derived from the standard model, on the ground of physical consideration [SLS+95]. In this model, additional mathematical constraints consistent with the physics of turbulent flows are enforced (*e.g.*, the positivity of the normal Reynolds stress). The performances of this model have been found to improve those of the standard k-ϵ model for several classes of flows, such as jets, mixing layers, boundary layers and separated flows. This aspect, together with its relatively low complexity and computational cost, makes it one of the most employed turbulence models.

k-ω model

Another popular two-equation model is the k-ω model introduced by, Wilcox [Wil93]. Here, a transport equation for the turbulence frequency $\omega = \epsilon/(C_\mu k)$ is added to that for the specific turbulent kinetic energy k:

$$\rho\partial_t k + \rho\bar{\boldsymbol{u}} \cdot \boldsymbol{\nabla} k = \boldsymbol{\nabla} \cdot [(\mu + \mu_T/\sigma_k)\boldsymbol{\nabla} k] + G_k - Y_k, \tag{1.31}$$

$$\rho\partial_t \omega + \rho\bar{\boldsymbol{u}} \cdot \boldsymbol{\nabla} \omega = \boldsymbol{\nabla} \cdot [(\mu + \mu_T/\sigma_\omega)\boldsymbol{\nabla} \omega] + G_\omega - Y_\omega, \tag{1.32}$$

where, G_k and G_ω represent the production terms for k and ω, respectively; σ_k and σ_ω are constants, while Y_k and Y_ω are the dissipation terms. Again, the form assumed by these terms is determined basing on empirical considerations.

After solving equations (1.31) and (1.32), the eddy viscosity can be computed as

$$\mu_T = \frac{\rho k}{\omega}, \tag{1.33}$$

Reynolds Stress model

The main limitation of turbulence models based on the Boussinesq hypothesis is that they assume the eddy viscosity to be an isotropic scalar quantity. In situations in which the anisotropy of the turbulent stresses has a dominant effect on the mean flow, one needs to solve transport equations for each component of the Reynolds stress tensor. This is done in the so-called *Reynolds Stress model* [LRR75]. In three dimensional problems, six equations (one for each turbulent stress tensor component) are added to the mean flow equations. An additional scale-determining equation (normally for ϵ) is also required. Despite its higher accuracy, this model is still less used than two-equation models, mostly because of its excessive computational cost.

For the numerical simulations that will be described in Chapters 3, 4 and 5, the realizable $k-\varepsilon$ model has been adopted. This choice has been based on a validation activity [Mol04] carried out for external flows around geometries similar to the ones considered in the present work. However, the calibration of turbulence models still deserves further investigation. In particular, a suitable tuning of the several model parameters for a specific class of problems requires an extensive validation work based on experimental measurements.

1.3.2 Laminar-turbulent transition

In presence of slender bodies aligned with the asymptotic velocity, as in the case of most naval engineering applications, a non negligible part of the flow field around the body can be laminar, and its simulation is crucial to obtain accurate predictions of drag coefficients.

In the context of RANS solvers, the development of numerical models able to predict accurately the laminar-turbulent transition is still an area of active investigation. Most of the RANS solver available on the market do not offer the possibility to simulate transition. In this respect, some encouraging results on transition models have been presented in [MLL$^+$04, LML$^+$04].

A possible workaround to this limitation consists in defining a *laminar zone* around specific region of the body, where the turbulent eddy viscosity is neglected in the averaged Navier–Stokes equations and turbulent quantities are only transported, having dropped from their equations all the source terms. This approach is possible only if an estimate location of the laminar to turbulent transition is known, based on experimental measurements. Clearly, the result of the simulation will depend on the quality of the prediction for the laminar zone position and extent. In Chapter 3, we will present a set of numerical simulations where this approach is adopted.

1.3.3 Wall treatment

In presence of a solid body, the mean velocity field has to satisfy the no-slip condition at the wall. Moreover, the turbulent behaviour of the flow is highly affected by the presence of the wall.

We consider a non-dimensional velocity scale u_τ, known as *friction velocity*, and a non-dimensional wall distance y^+ defined by

$$u_\tau = \sqrt{|\tau_w|/\rho},$$
$$y^+ = \rho y u_\tau / \mu,$$

where τ_w is the wall shear stress and y is the wall distance.

Experimental analyses have shown that the near-wall region of turbulent boundary layers can be subdivided into three main sub-regions (see Fig. 1.3): the so-called *viscous sublayer* ($y^+ < 5$), where turbulent fluctuations are damped by molecular viscosity, a intermediate *buffer layer* ($5 < y^+ < 60$), where viscous and turbulent effects are equally important, and a fully turbulent layer ($y^+ > 60$), where the dependence of the tangential mean velocity on the wall distance is described by the so-called *logarithmic law*:

$$\frac{u}{u_\tau} = 2.5 \log\left(\frac{\rho u_\tau y}{\mu}\right) + 5.45,$$

which has been obtained experimentally for the turbulent boundary layer on a flat plate. On the other hand, in the viscous sublayer, the mean velocity depends linearly on y^+ (see, *e.g.*, [Sch68]).

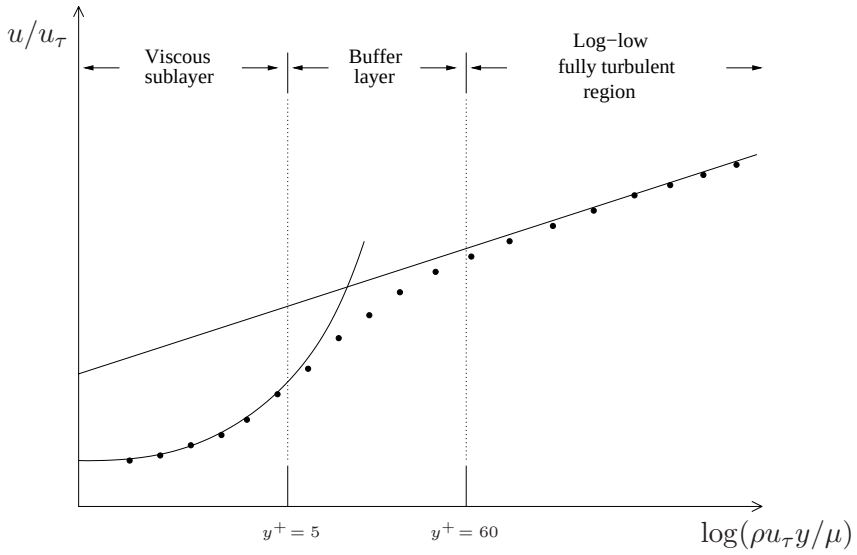


Figure 1.3: Near-wall region in turbulent boundary layers.

It is well-known that solving the turbulence equations described above through the near-wall region and applying the no-slip condition at the wall yields unsatisfactory results. A possible way to overcome this deficiency is to introduce the so-called *wall functions* to model the near-wall region. Wall functions use empirical laws to circumvent the inability of turbulence models to predict a velocity profile near a wall. With these laws it is possible to express the mean velocity parallel to the wall and turbulence quantities outside the viscous sublayer in terms of the distance to the wall and wall conditions such as wall shear stress and pressure gradient. Indeed, the wall functions are used to provide near-wall boundary conditions for the momentum and turbulence transport equations, rather than conditions at the wall itself, so that the viscous sublayer does not have to be resolved [LS72].

The use of wall functions is limited to case with moderate pressure gradients and in absence of flow separations. An alternative approach is based on the correction of the turbulence models in order to account for the low Reynolds number effects (see, *e.g.*, [Wol69]). In this case, the computational grid has to resolve the near-wall region, including the viscous sublayer. However, for high Reynolds number flows this approach can turn out to be excessively expensive from the computational point of view. We refer to [Flu03] for a detailed description of the wall treatment adopted in the simulation presented in Chapter 3, 4 and 5.

Chapter 2

Numerical Discretization based on Finite Volumes

In this chapter, we describe the numerical schemes adopted for the discrete solution of the mathematical problems introduced in Chapter 1. The Volume of Fluid (VOF) method, described in Section 1.2.2, is used for tracking the free-surface, while turbulent effects are taken into account by means of the realizable $k-\epsilon$ turbulence model introduced in Section 1.3. The numerical model described in this chapter is the one implemented in the commercial software Fluent^{TM}, a general purpose CFD code that has been used to address a wide range of problems in fluid mechanics. One of the objectives of this thesis is to investigate the behaviour of this solver for the solution of ship hydrodynamics problems.

The spatial discretization is based on a cell-centered finite-volume approach on general (structured or unstructured) grids. The governing equations are integrated over control volumes (identified with the elements of the computational grid) yielding discrete equations that conserve each quantity on a control-volume base. The solution of the Navier–Stokes system is based on a SIMPLE like strategy [Pat80], where the pressure computation is decoupled from the computation of the velocity field. The equation associated to the VOF method and to the turbulence model, which are also discretized using finite volumes, are decoupled from the Navier–Stokes system and are solved sequentially at each time iteration.

2.1 Integral form of the flow equations

In view of the finite volume spatial discretization, we derive the integral form of the governing equations introduced in Chapter 1.

We first consider the Reynolds averaged Navier–Stokes equations (1.23)-(1.24), with the Reynolds stress tensor modelled according to Boussinesq hypothesis (1.25). For an arbitrary control volume $c \subset \Omega$, using Gauss theorem, equations (1.23)-(1.24) can be rewritten in integral form, as follows:

$$\int_c \partial_t(\rho \boldsymbol{u}) dV + \oint_{\partial c} \rho\, \boldsymbol{u} \otimes \boldsymbol{u} \cdot \boldsymbol{n}\, d\Gamma - \oint_{\partial c} \boldsymbol{T}(\boldsymbol{u}, p) \cdot \boldsymbol{n}\, d\Gamma = \int_c \rho \boldsymbol{g} dV, \tag{2.1}$$

$$\oint_{\partial c} \rho\, \boldsymbol{u} \cdot \boldsymbol{n}\, d\Gamma = 0, \tag{2.2}$$

where $\boldsymbol{n}$ is the outward unit normal vector on the control volume boundary ∂c, $\boldsymbol{T}(\boldsymbol{u},p) = (\mu+\mu_T)(\boldsymbol{\nabla u}+\boldsymbol{\nabla u}^T) - p^*\boldsymbol{I}$ is the effective stress tensor and $p^* = p + 2/3\rho k$ is the modified pressure (according to (1.25)). With respect to notations in equations (1.23)-(1.24), the overbar expressing the mean value of velocity and pressure has been dropped for ease of reading.

Analogously, we can derive the integral form of the VOF equation (1.16) for the volume fraction ψ:

$$\int_c \partial_t \psi dV + \oint_{\partial c} \psi\, \boldsymbol{u}\cdot\boldsymbol{n}\, d\Gamma = 0, \tag{2.3}$$

and of the turbulence equations (1.28)-(1.29) for k and ϵ:

$$\int_c \partial_t(\rho k) dV + \oint_{\partial c} \rho\, k\boldsymbol{u}\cdot\boldsymbol{n}\, d\Gamma - \oint_{\partial c} \mu_k \boldsymbol{\nabla} k\cdot\boldsymbol{n}\, d\Gamma = \int_c \boldsymbol{F}_k dV, \tag{2.4}$$

$$\int_c \partial_t(\rho \epsilon) dV + \oint_{\partial c} \rho\, \epsilon\boldsymbol{u}\cdot\boldsymbol{n}\, d\Gamma - \oint_{\partial c} \mu_\epsilon \boldsymbol{\nabla} \epsilon\cdot\boldsymbol{n}\, d\Gamma = \int_c \boldsymbol{F}_\epsilon dV, \tag{2.5}$$

where $\mu_k = (\mu + \mu_T/\sigma_k)$, $\mu_\epsilon = (\mu + \mu_T/\sigma_\epsilon)$ and the production and dissipation terms for k and ϵ have been grouped, respectively, in source terms F_k and F_ϵ.

2.1.1 Finite volume discretization

For the reader's convenience, let us recall the general principles behind finite volumes introducing the discretization of a simple steady advection-diffusion equation for a scalar quantity ϕ:

$$-\boldsymbol{\nabla}\cdot(\mu\,\boldsymbol{\nabla}\phi - \rho\,\boldsymbol{u}\,\phi) = S, \qquad \text{in } \Omega, \tag{2.6}$$

where ρ is a given positive scalar function, $\boldsymbol{u}$ is a given convective field, μ is a non-negative diffusion coefficient and S is a source term per unit volume. For an arbitrary control volume c, we have

$$\oint_{\partial c} \rho\,\phi\,\boldsymbol{u}\cdot\boldsymbol{n}\, d\Gamma = \oint_{\partial c} \mu\,\boldsymbol{\nabla}\phi\cdot\boldsymbol{n}\, d\Gamma + \int_c S\, dV. \tag{2.7}$$

The discretization of equation (2.7) on a given cell c (now cell stands for control volume) reads

$$\sum_f^{N_{\text{faces}}} \rho_f\,\phi_f\,\boldsymbol{u}_f\cdot\boldsymbol{n}_f\,|A_f| = \sum_f^{N_{\text{faces}}} \mu_f\,(\boldsymbol{\nabla}\phi)_f\cdot\boldsymbol{n}_f\,|A_f| + S\,|V_c|, \tag{2.8}$$

where the index f represents quantities evaluated on a given face f, N_{faces} is the number of faces of the cell contour, $|A_f|$ is the area of face f and $|V_c|$ is the measure (volume) of cell c. Within a given cell, all the quantities are supposed to be constant and their constant value is identified with the pointwise value at the cell center of gravity. Consequently, the face values need to be suitably defined.

Approximation of the convective fluxes

The convective fluxes are obtained by an upwind interpolation from the adjacent cell center values. In our scheme a second-order upwind scheme based on a multidimensional linear reconstruction

approach [Bar89] has been used. The value at the cell face ϕ_f is obtained through a Taylor series expansion of the cell-centered solution about the upwind cell centroid $\boldsymbol{x}_{uc}$, namely

$$\phi_f = \phi_{uc} + \boldsymbol{\nabla}\phi_{uc} \cdot \boldsymbol{r},$$

where ϕ_{uc} and $\boldsymbol{\nabla}\phi_{uc}$ are the cell-centered value and its gradient in the upstream cell, and $\boldsymbol{r}$ is the displacement vector from the upstream cell centroid $\boldsymbol{x}_{uc}$ to the face centroid $\boldsymbol{x}_f$ (see Fig. 2.1). The gradient $\boldsymbol{\nabla}\phi_{uc}$ is computed, applying the divergence theorem in the upwind cell, as follows

$$\boldsymbol{\nabla}\phi_{uc} = \frac{1}{|V_{uc}|} \sum_{f}^{N_{\text{faces}}} \widetilde{\phi}_f |A_f|, \tag{2.9}$$

where $|V_{uc}|$ is the volume of the upwind cell and the face values $\widetilde{\phi}_f$ are computed by averaging ϕ from the two cells adjacent to the face.

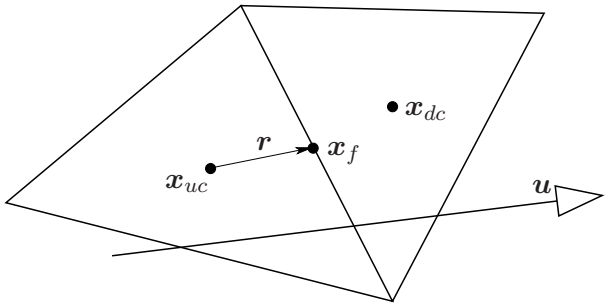


Figure 2.1: Second order upwinding for the convective fluxes.

Approximation of the diffusive fluxes

For the diffusion terms in equation (2.8) a second-order accurate central scheme is used. The face value ϕ_f is defined as follows:

$$\phi_f = \frac{1}{2}(\phi_0 + \phi_1) + \frac{1}{2}(\boldsymbol{\nabla}\phi_0 \cdot \boldsymbol{r}_0 + \boldsymbol{\nabla}\phi_0 \cdot \boldsymbol{r}_1), \tag{2.10}$$

where the indexes 0 and 1 refer to the cells that share face f, $\boldsymbol{\nabla}\phi_0$ and $\boldsymbol{\nabla}\phi_1$ are the gradients computed by (2.9) at cells 0 and 1, respectively, and $\boldsymbol{r}$ is the vector directed from the cell centroid toward the face centroid.

2.2 Discretization of the flow equations

The spatial discretization of the Navier–Stokes equations (2.1)-(2.2), as well as the one of VOF and turbulence equations (2.3), (2.4) and (2.5), is accomplished by first linearizing the equation at hand, then proceeding like for equation (2.6).

For every cell, equation (2.8) can be written as an algebraic equation of the following form:

$$a_c \phi_c = \sum_{nb} a_{nb} \phi_{nb} + b, \tag{2.11}$$

where c is the cell index while the index nb ranges over the neighbor cells, and a_c and a_{nb} are suitable coefficients. This results in a set of algebraic equations with sparse coefficient matrix for the vector $\{\psi_c\}_{c=1,\cdots,N_c}$ (with N_c denoting the number of cells).

2.2.1 SIMPLE algorithm

As previously mentioned, the solution algorithm is based on the SIMPLE method, a pressure-correction iterative numerical scheme for incompressible flows.

To describe this algorithm, we consider the steady Navier–Stokes equations in integral form:

$$\oint_{\partial c} \rho\, \boldsymbol{u} \otimes \boldsymbol{u} \cdot \boldsymbol{n}\, d\Gamma - \oint_{\partial c} \boldsymbol{T}(\boldsymbol{u}, p) \cdot \boldsymbol{n}\, d\Gamma = \int_c \rho \boldsymbol{g} dV, \tag{2.12}$$

$$\oint_{\partial c} \rho\, \boldsymbol{u} \cdot \boldsymbol{n}\, d\Gamma = 0. \tag{2.13}$$

After linearization, the finite volume discretization of the x-component of equation (2.12) yields an algebraic problem of the following form:

$$a_c u_c = \sum_{nb} a_{nb} u_{nb} + \sum_{f}^{N_{\text{faces}}} p_f\, \boldsymbol{n}_f \cdot \boldsymbol{i}\, |A_f| \tag{2.14}$$

where $\boldsymbol{i}$ is the unitary vector in the x-direction and the u variable indicates the first component of the vector field $\boldsymbol{u}$. Similar equations hold for the y and z components, however on the latter a source term $-\rho g |V_c|$ shows up in the right hand side.

The basic ideas underlying the SIMPLE method are more easily explained considering the algebraic system arising from the discretization of equations (2.12)-(2.13), that reads:

$$\begin{pmatrix} C & G \\ G^T & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \tag{2.15}$$

where U and P are the algebraic vectors corresponding to the velocity components and the pressure, respectively; C is the matrix corresponding to the viscous, convective and mass terms in (2.12), G and G^T are the matrices corresponding to the gradient and divergence operators, respectively. We denote with D the diagonal of matrix C. Given an initial estimate of the pressure P^*, the SIMPLE method is described by the following iterative algorithm:

1. solve $C\,\tilde{U} = b_1 - G\,P^*$;
2. solve $R\,P' = b_2 - G^T\,\tilde{U}$, with $R = -G^T\,D^{-1}\,G$;
3. compute $U = \tilde{U} - D^{-1}\,G\,P'$ and set $P = P^* + P'$;
4. if not converged, set $P^* = P$ and go to 1.

Note that, in steps 2 and 3, D^{-1} is used instead of the true inverse C^{-1} to reduce computational complexity.

In the framework of the previous finite volume discretization, each iteration of the SIMPLE method consists in the following steps:

1. We solve the momentum equations using the value of pressure p^* from previous iteration and velocity, in order to compute an intermediate velocity field $\tilde{\boldsymbol{u}}$ which is not divergence free. Its x-component is:

$$a_c \tilde{u}_c = \sum_{nb} a_{nb} \tilde{u}_{nb} + \sum_{f}^{N_{\text{faces}}} p_f^* \, \boldsymbol{n} \cdot \boldsymbol{i} \, |A_f|. \tag{2.16}$$

2. We denote the new pressure $p = p^* + p'$, p' being a correction. By imposing the mass conservation (2.13), we find $b_2 = G^T U = G^T C^{-1} (b_1 - G P) = G^T \tilde{U} + R P'$. With the help of a little algebra, this yields the following (Poisson-type) equation for the pressure correction:

$$a_c p'_c = \sum_{nb} a_{nb} p'_{nb} + b_2. \tag{2.17}$$

   The right hand side term b_2 is the net flow rate into the cell, defined as

$$b_2 = \sum_{f}^{N_{\text{faces}}} J_f \, A_f, \tag{2.18}$$

   where the face flux J_f is computed using a momentum-weighted average based on the a_c coefficients from equation (2.14). This is necessary to prevent the checkerboarding that occurs if linear interpolation of cell-centered velocities is used in combination with a collocated scheme (velocity and pressure associated to the same position), as shown by Rhie and Chow [RL83].

3. Correcting the cell pressure and face flux

$$p = p^* + \alpha \, p', \tag{2.19}$$

$$J_f = J_f^* + \beta \, d_f (p'_{c0} - p'_{c1}), \tag{2.20}$$

   where α and β are positive dumping factors between 0 and 1 (which help convergence to the steady state), p'_{c0} and p'_{c1} are the pressure corrections within the two cells on either side of face f and d_f is a function of the average of the momentum equation a_c coefficients for the cells on either side of face f.

The iteration loop is stopped when the convergence criteria, based on the scaled residuals of each equation, are fulfilled.

The solution of the three linear algebraic systems in (2.16) (one for each velocity component) and of the one for the pressure correction in (2.17), as well as of those obtained from the discretization of the VOF and turbulence equations, is accomplished by an algebraic multigrid method [HD86] with Gauss-Seidel smoothing.

2.2.2 Time discretization

The temporal discretization of equations (2.1), (2.3), (2.4) and (2.5) is based on a second-order implicit scheme. We consider a uniform decomposition of the time interval $[0, T]$ into N subintervals. Denoting with $\Delta t = T/N$ the time step, we use the index n to denote variables at time $t^n = n \, \Delta t$, with $n = 0, \ldots, N$.

Let us consider the time evolution of a generic variable ϕ given by

$$\partial_t \phi = F(\phi), \tag{2.21}$$

where F represents the differential operator containing any spatial discretization. A second-order accurate discretization of equation (2.21) is obtained using the Backward Difference Formula (BDF2), that reads:

$$\frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\Delta t} = F(\phi^{n+1}). \tag{2.22}$$

A second-order accurate solution of the implicit equation (2.22) is obtained by setting $\phi_0^{n+1} = \phi^n$ and iterating the equation

$$\phi_{i+1}^{n+1} = \frac{4}{3}\phi^n - \frac{1}{3}\phi^{n-1} + \frac{2}{3}\Delta t F(\phi_i^{n+1}),$$

until convergence on the residuals is reached.

2.3 Overview of the solution algorithm

In conclusion, the numerical algorithm can be summarized as follows. Let us consider $\boldsymbol{U}^n$, $\boldsymbol{P}^n$, $\boldsymbol{\Psi}^n$, $\boldsymbol{K}^n$ and $\boldsymbol{E}^n$, defined as the vectors containing the discrete values of, respectively, $\boldsymbol{u}$, p, ψ, k and ϵ in each volume control, at time $T = n\Delta t$. We introduce the global discrete solution $\boldsymbol{Y}^n$ defined as:

$$\boldsymbol{Y}^n = \left\{ \begin{array}{c} \boldsymbol{U}^n \\ \boldsymbol{P}^n \\ \boldsymbol{\Psi}^n \\ \boldsymbol{K}^n \\ \boldsymbol{E}^n \end{array} \right\}$$

Given $\boldsymbol{Y}^n$, the global discrete solution at time $(n+1)\Delta t$ is computed by setting $\boldsymbol{Y}_0 = \boldsymbol{Y}^n$ and looping over the following successive steps, for $i \geq 0$:

1. update the variables $\boldsymbol{Y}_{i+1} = \boldsymbol{Y}_i$
2. solve the momentum equation (2.16) for each velocity component and compute the intermediate velocity $\bar{\boldsymbol{U}}_{i+1}$;
3. solve the pressure correction equation (2.17) and compute pressure $\boldsymbol{P}_{i+1}$ and final velocity $\boldsymbol{U}_{i+1}$;
4. solve the VOF equation and compute $\boldsymbol{\Psi}_{i+1}$;
5. solve the $k-\epsilon$ equations and compute $\boldsymbol{K}_{i+1}$ and $\boldsymbol{E}_{i+1}$;
6. if convergence on the residuals is reached for all the equations, set $\boldsymbol{Y}^{n+1} = \boldsymbol{Y}_{i+1}$ and advance in time, otherwise set $\boldsymbol{Y}_i = \boldsymbol{Y}_{i+1}$ and go to step 1.

A flowchart representation of the algorithm is displayed in Fig. 2.2.

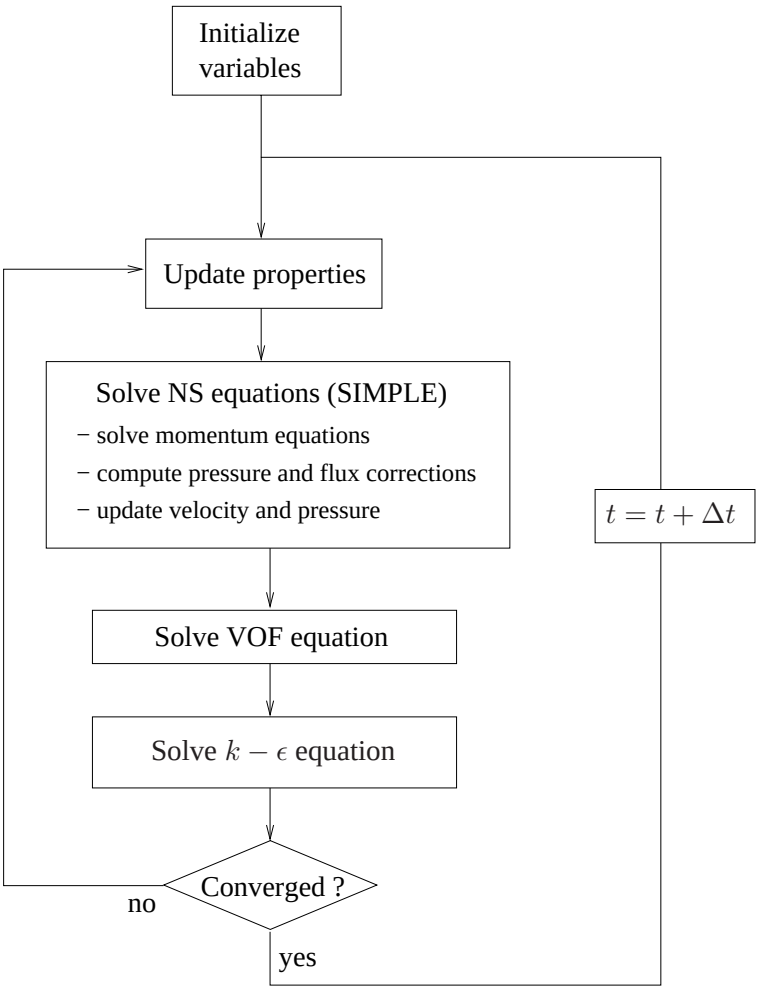


Figure 2.2: Overview of the solution algorithm.

2.4 Parallel approach

The numerical simulations that will be presented in the following chapters have been carried out with the numerical scheme introduced here on computational grids of large sizes (up to 6 million elements). In a cell-centered finite volume approach, the solution of the Reynolds Averaged Navier–Stokes equations coupled with the Volume of Fluid method and the $k-\epsilon$ turbulence model involves seven variables (three velocity components, pressure, volume fraction and the turbulence quantities k and ϵ) which are defined as piecewise constant on each grid element. This means that the discrete problems solved in the larger simulations performed involve up to 42 millions of unknown.

The solution of such large scale problems demands for a parellel implementation of the numerical method. In the code we have used (Fluent), parallelism is based on a domain decomposition technique. The computational grid is decomposed in a number of *partitions* corresponding to the number of processors on which the simulation will be carried out. In order to minimize communication between processors, optimal surface-volume ratios are achieved using the Metis graph-partitioning scheme. The number of processors to use should be chosen accordingly to the

size of the simulation.

Once the grid has been partitioned, a host process assigns each partition to one processor. Each processor simultaneously executes the same operation on its own data and is virtually connected to the other processors through a *communicator*. The communicator, which is based on a message pass6ing interface (MPI) system, is devoted to the exchange of boundary information between partitions as well as to keep the synchronization between processors.

The large scale simulations have been carried out on two different computer systems:

- an SGI Origin 3800 central computer system with 128 MIPS R14000 (500 MHz) processors and 64 GB of shared memory;
- a Linux Cluster with 20 Intel Xeon (2.8 GHz) two-processor nodes with 3 GB of memory on each node and a communication system based on a 24 ports Gigabit Ethernet switch.

A brief discussion on the parallel performance of the code for the simulations performed will be presented in Chapter 3. For a detailed analysis of the parallel performance of the code on different hardware architectures, we refer to [KTP00].

Chapter 3

Numerical Simulations for Yacht Design

3.1 Introduction

The America's Cup is a yachting race first run in 1851 and whose prize represents one of the world's oldest sporting trophies. America's Cup yacht races are fiercely competitive. Even after a race of several hours, just seconds can separate two teams at the finish line. Unlike aerospace or automotive industrial applications, for which vast safety margins must be guaranteed, America's Cup competition demands the utmost performance. Design teams must perform extensive analyses to ensure that their racing yachts can pursue victory without slipping over the edge into disaster.

The 31st America's Cup series held during spring 2003 in Auckland, New Zealand, was won by the swiss syndicate Alinghi. In July 2001, the Alinghi Team named the Federal Institute of Technology in Lausanne (EPFL) as its official scientific advisor. The partnership involved different research areas: the study of high-performance composite materials, the development of new material processing methods, the analysis of the hydrodynamic and aerodynamic flows around the yacht by means of experimental tests and numerical simulations.

In this chapter, we will describe the numerical simulations carried out in the framework of the collaboration with the Alinghi Team and based on the numerical scheme introduced in Chapter 2. An overview on the role that the CFD analyses have played in the global design process will be given. Moreover, we will report the main results obtained in the different design research areas that have been investigated. Some of the results presented here have been already published in [CPS03] and [PQ04].

3.2 An overview on America's Cup yacht design

An America's Cup yacht is a very sophisticated system that should operate optimally in a wide range of sailing conditions. The different components (over and under the water surface) that compose a sailing yacht interact one another through several complex relations. The design of an America's Cup yacht must account for this complexity and requires to set up suitable (experimental and numerical) tools able to describe as accurately as possible the system, in order to achieve an optimal configuration. In Fig. 3.1, the different components on the emerged and submerged parts of a typical America's Cup yacht are presented.

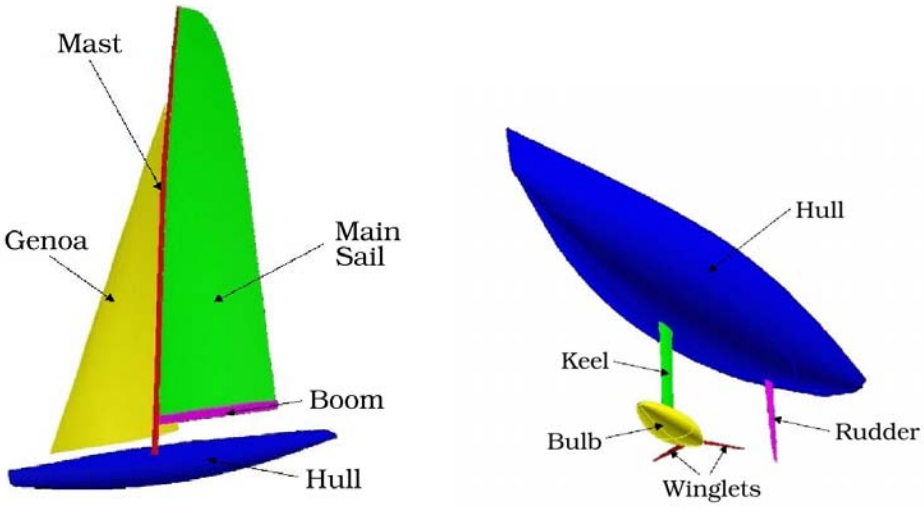


Figure 3.1: Different components on the emerged (left) and submerged (right) parts of an America's Cup yacht.

To give an idea of the complex interactions that should be taken into account in the design process, let us consider a simple example. One way to reduce the viscous resistance on the hull, that is the force in the course direction given by frictional effects, is reducing its wetted surface. This can be accomplished by reducing its beam (width), keeping the same boat length. A reduction of beam decrease the heeling stability (*i.e.* the stability around the longitudinal centerline of the boat) and, consequently, the forces on the sails. We can see how a single change involves a sequence of effects on different areas of the design process.

Moreover, an America's cup yacht is constrained by the rules of the International America's Cup Class (IACC). This class was first introduced in 1992 and continuously evolves from one edition to the next. Severe restrictions are imposed on a number of design factors, such as geometrical dimensions (depth, displacement, sail area), flow control devices (*e.g.* number of underwater moving surfaces) and materials. In particular, one rule has played a crucial role in the evolution to the current America's Cup configuration. This rule, known as the "Formula", is based on a relation between boat length L_b, sail area A_s and displacement D:

$$\frac{L_b + 1.25\sqrt{A_s} - 9.8\sqrt[3]{D}}{0.686} \leq 24 \text{ m} \tag{3.1}$$

This constraint involves, for example, that to have a longer boat one should lower the sail area or increase the displacement.

In America's Cup match race, two buoys at a distance of 18.55 nautical miles are positioned in the wind direction. Three laps between the two buoys have to be completed, resulting in three upwind and three downwind legs (see Fig. 3.2). Upwind and downwind sailing call for different sailing techniques and the design of the boat should account for the conflicting requirements coming from the two regimes. For the sail rig, this problem is overcome through the use of different sets of sails (main and genoa for upwind sailing, main and spinnaker/gennaker for downwind sailing). On the other hand, in the underwater part, the possible changes during the race are restricted to the trimming of rudder and keel trim tab. Yacht appendices have to be designed to perform both in downwind sailing, where minimal drag should be attained, and in upwind sailing, where they have to resist the forces and moments generated by the sails.

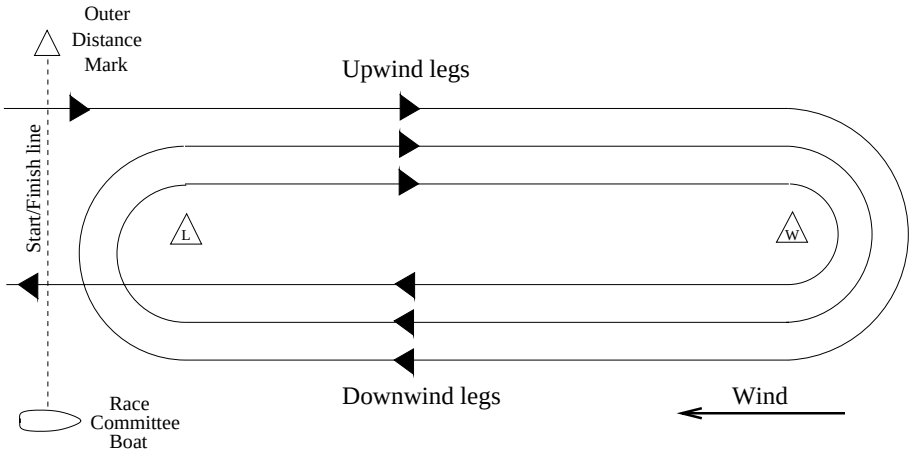


Figure 3.2: America's Cup race course.

The standard approach adopted in the America's Cup design teams to evaluate whether a design change (and all the other design modifications that this change implies) is globally advantageous, is based on the use of a Velocity Prediction Program (VPP). VPP programs can be used to estimate the boat speed (and, in certain cases, the boat attitude) for any prescribed wind condition and sailing angle β_{TW} (the angle between the centerline of the boat and the wind direction). A numerical prediction of boat speed and attitude can be obtained by modeling the balance between the aerodynamic and hydrodynamic forces acting on the boat. A diagram representing the hydrodynamic and aerodynamic force and moment components acting in the water plane is presented in Fig. 3.3.

On the water plane, a steady sailing condition is obtained imposing two force balances in x direction (aligned with the boat velocity) and y direction (normal to x on the water plane) and a heeling moment balance around the centerline of the boat:

$$\begin{aligned} \boldsymbol{D}_h + \boldsymbol{T}_a &= 0, \\ \boldsymbol{S}_h + \boldsymbol{S}_a &= 0, \\ \boldsymbol{M}_h + \boldsymbol{M}_a &= 0, \end{aligned} \tag{3.2}$$

where $\boldsymbol{D}_h$ is the hydrodynamic drag (along the course direction), $\boldsymbol{T}_a$ is the aerodynamic thrust, $\boldsymbol{S}_h$ is the hydrodynamic side force perpendicular to the course, $\boldsymbol{S}_a$ is the aerodynamic side force, $\boldsymbol{M}_h$ and $\boldsymbol{M}_a$ are, respectively, the hydromechanical righting moment and the aerodynamic heeling moment around the boat meanline. The angle β_Y between the course direction and the boat centerline is called *yaw* angle. The aerodynamic thrust and side force can be seen as a decomposition in the reference system aligned with the course direction of the aerodynamic lift and drag which are defined on a reference system aligned with the apparent wind direction (see Fig. 3.3). Similar balance equations can be obtained for the other degrees of freedom, namely the vertical displacement and pitching and yawing rotations.

In a VPP program, all the terms in system (3.2) are modeled as functions of boat speed, heel angle and yaw angle. Suitable correlation between the degrees of freedom of the system and the different force components can be obtain based on different sources of information: experimental results, theoretical predictions and numerical simulations. For a detailed presentation of Velocity Prediction Programs, we refer to [Lar90, Mil93].

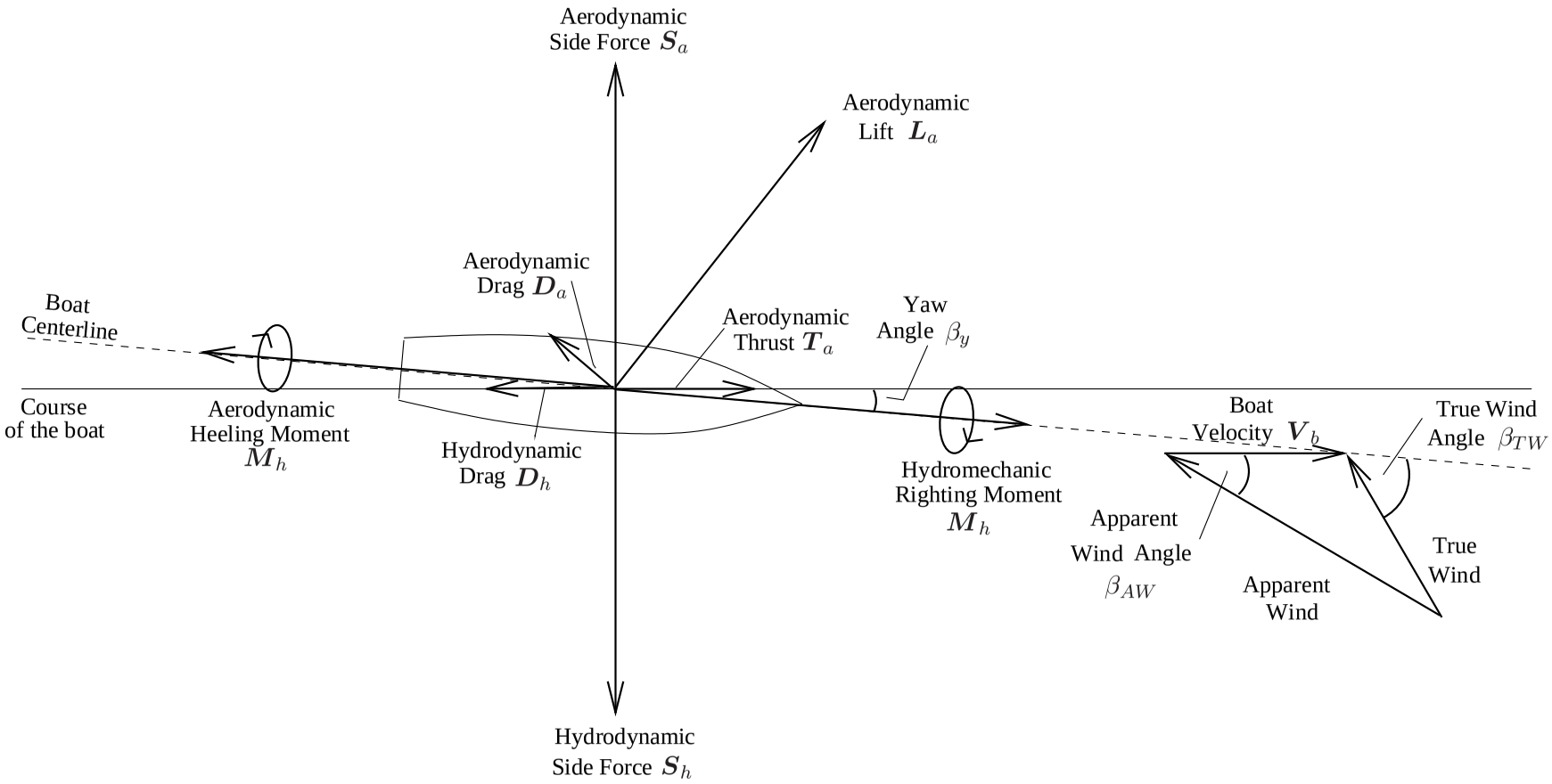


Figure 3.3: Forces and moments on the water plane.

The role of advanced Computational Fluid Dynamics is to supply accurate estimate of the forces acting on the boat in different sailing condition in order to improve the reliability of the prediction of the overall performance associated to a given design configuration.

In this chapter, we focus our attention on the prediction of some of the force components contributing to the global balance (3.2). We will present the numerical simulations that have been carried out to estimate the forces acting on the yacht appendages in different sailing conditions and to investigate the role of different design parameters. Some results concerning the aerodynamic flow around sails are also presented.

3.3 The mathematical problem and its complexity

One trend that helps explain the importance of technology and testing in the America's Cup is, ironically, uniformity. A decade ago, racing teams experimented with a variety of hull and keel shapes; today, geometries have converged toward standardized shapes, and the smallest details make the difference. Jerome Milgram, professor of Ocean Engineering at MIT and veteran of a dozen America's Cup contests, presented in [Mil98] a clear analysis of the impact that small differences in design can induce on the overall performance. In Fig. 3.4, we report a graphics extracted from [Mil98] showing the effect on race time given by 1% change in total drag. The time differentials, on the range of wind speeds considered, varies from 18 to 64 seconds and such deltas can easily make the difference between a victory and a defeat.

Numerical predictions with such a level of accuracy on absolute force estimates are still difficult to obtain (at least for this class of problems). However, when different design configurations are considered, it is possible to estimate, on a relative basis, which one performs better.

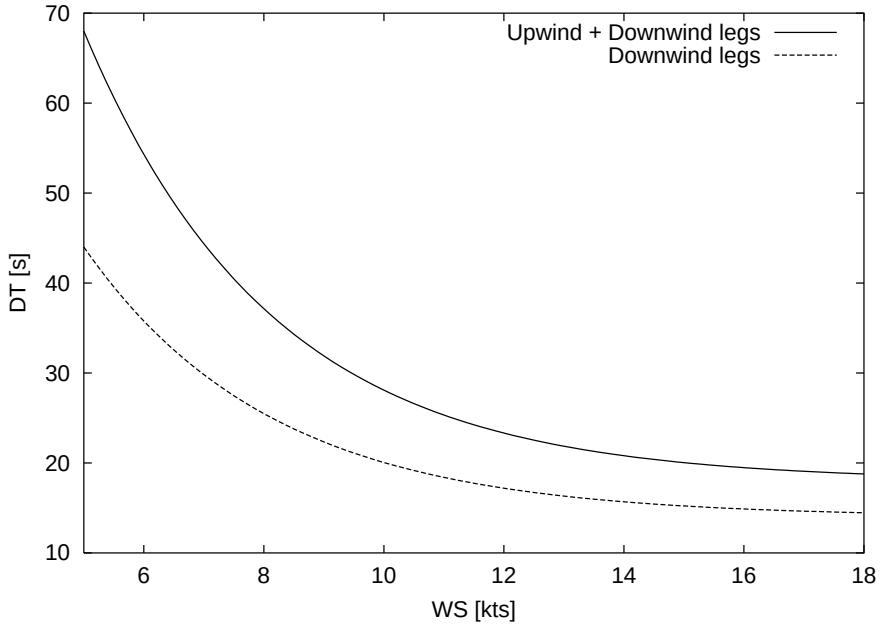


Figure 3.4: Time differentials (DT) at different wind speed (WS), on the entire course and downwind legs only, corresponding to a 1% change in resistance (extracted from [Mil98]).

3.3.1 Devising hierarchical models to reduce complexity

A complete model describing the physical behaviour of a yacht in racing condition should be based on the coupling between Navier–Stokes equations with free-surface and the equations for structure deformation. Such a model has a prohibitive complexity. This complexity and the strict time constraints have prompted to the definition of a hierarchy of mathematical models (to be integrated in the overall design process) characterized by different levels of accuracy, complexity, computational cost, ease of use and acceptance within the design community.

Solving the complete problem by neglecting the deformability of the structure is the first assumption that can be made in order to reduce the complexity. Moreover, since the main subjects of investigation, in the present work, have been the design of appendages and the analysis of the flow around the sails, a steady water plane was considered. Free-surface flow employing the technique introduced in Section 1.2 will be considered in Chapters 4 and 5.

Although simplified by having neglected the structural deformations and the free-surface dynamics, yet the numerical simulations have a tremendous computational complexity, since they require:

- the translation of a new design concept (usually provided by the design team) into its geometrical description;
- the generation of a suitable three dimensional grid capable of capturing all the relevant flow features;
- the solution of the RANS equations by parallel computers in the computational domain;
- the post-processing of the results.

3.3.2 Further simplified models

The results obtained from our simulations have been analysed by the design team. They were compared and combined with those obtained by experimental analyses (in the wind tunnel or in watertank) or by the computer using even more simplified models which can be run in real time on a personal computer. These simplified methods are usually based on potential flow (where fluid viscosity is neglected as well as vorticity creation and propagation). They can provide fast valuable information (surface pressure and global forces) for non-separated flows such as the hydrodynamic flow around boat appendages and the aerodynamic flow around upwind sailing configurations. The results obtained by RANS models have been integrated into the overall design process, in particular with the results obtained from different computational tools mainly used in the Alinghi design team. They include:

- a free-surface potential flow panel code (Splash^{TM}) for aero/hydrodynamic design and analysis, to provide the free-surface location.
- a 2D boundary layer solver (XFoil^{TM}) and a 3D boundary layer solver (3C3D^{TM}) for non-separated flows, to compute the location where the transition from laminar to turbulent flow regime does occur. This information is then used in the RANS solver to identify the subregions of the domain in which the turbulence model should be switched on;
- a potential flow panel code (Flow-MemBrain^{TM}) for fluid-structure simulations of sails, to create the deformed sails geometry used afterward in the RANS simulations.

3.4 Grid generation

The most onerous part of the simulation procedure in terms of manpower time and effort is the generation of the computational grid. In general, this process consists on three major steps:

- CAD geometry cleaning,
- surface meshing,
- volume meshing.

In the first step, the CAD representation of the geometry must be rigorously examined to ensure that it meets the constraints of the grid generation software in terms of surface tolerance and port/starboard symmetry (if applicable). CAD geometries that are unsuitable for CFD analysis (due to the presence of very small surfaces, gaps and/or overlaps) can be a major bottleneck in the grid generation process; communication between the engineers performing the CAD design and the grid generation is critical. Since most grid generation software has only limited flexibility for the repair and manipulation of the CAD object, streamlining and determination of the optimal settings in the CAD work is crucial.

Flexibility in the grid topology is beneficial and perhaps critical for treating complex geometries such as underwater region of an IACC boat. In general, when using unstructured mixed-element grids to resolve high-Reynolds-number viscous flow, the surfaces are meshed with triangles/quadrilaterals and layers of prisms/hexahedra are then extruded from the surface to produce a region to capture the boundary shear layer. In the final step, tetrahedrons are generated to fill the remaining volume of the computational domain.

When the surface curvature of critical components, such as keel and winglet leading edges, is very large, triangulation of the surface will lead to an unmanageable number of elements since the element aspect ratio must remain near unity. For these regions, a quadrilateral surface grid was used in this work since element aspect ratios could be increased by an order of magnitude. The short side of the quadrilateral followed the chordwise direction and the long side was placed spanwise. The final surface grid contained around 200,000 elements and utilized a combination of both triangular and quadrilateral elements to provide adequate surface resolution with a grid of realistic size for an overnight computation (see Fig. 3.5 for an example of surface grid on the complete boat).

After completion of the surface meshing, the volume containing the shear layer is created (see Fig. 3.6). The layer initial spacing, stretching, and thickness of the region are dictated by the Reynolds number, the spatial accuracy of the flow solver, the turbulence model, and the method used to resolve the inner region of the turbulent boundary layer.

In particular, the total thickness of the prismatic grid is chosen based on an estimation of the thickness of the turbulent boundary layer. Moreover, to satisfy the requirements on the mesh spacing in the wall region (see Section 1.3.3), empirical laws which describe the turbulent boundary layer on a flat plate can be used. For example, an estimation of y^+ can be obtained by the following relation [Sch68]:

$$y^+ = 0.172 \, \frac{y}{L} \mathrm{Re}^{0.9},$$

where y is the distance from the wall, L is a characteristic length and Re is the Reynolds number.

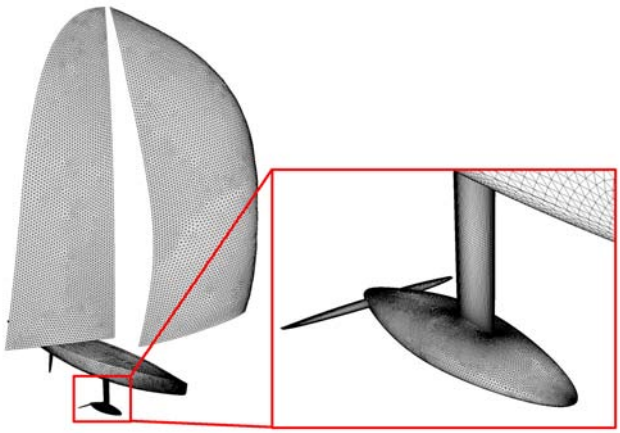


Figure 3.5: Surface grid for an IACC yacht.

The algorithm governing the above-mentioned grid extrusion process is controlled by a large number of parameters that enhance its stability and quality. Determining the combination that results in the desired initial layer spacing, number of layers and total thickness of the region can be very time consuming. In regions of high concavity such as at the winglet/bulb intersection, the growth process often fails and alternate parameters must be chosen.

For example, the normal growth of a prism layer in the keel-bulb or winglet-bulb junction regions becomes critical whenever the total height of the prismatic layer gets significantly larger than the local dimension of the surface cells, as shown with a two-dimensional sketch in Fig. 3.7. Indeed, for large thickness of the prismatic layer, the cells extruded in normal direction from the two intersecting surfaces overlap, eventually leading to the generation of (unacceptable) negative-volume cells. One solution to this problem is relaxing the normal constraint in the extrusion direction, growing the exterior layers non normally to the wall. A possible drawback of this

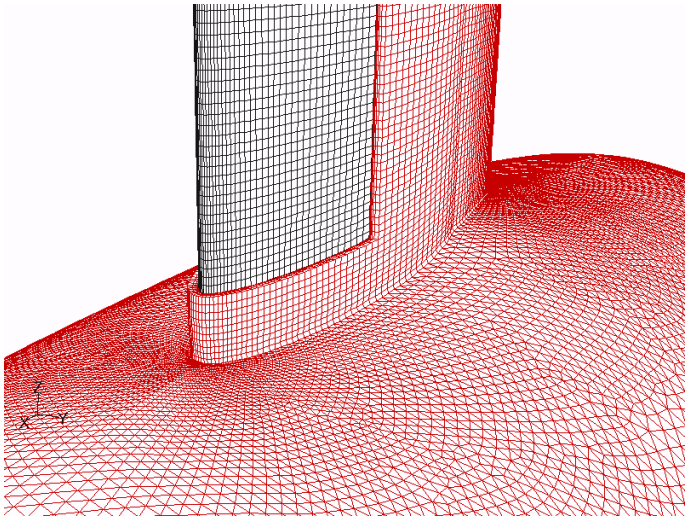


Figure 3.6: Boundary Layer prismatic grid on a keel-bulb configuration.

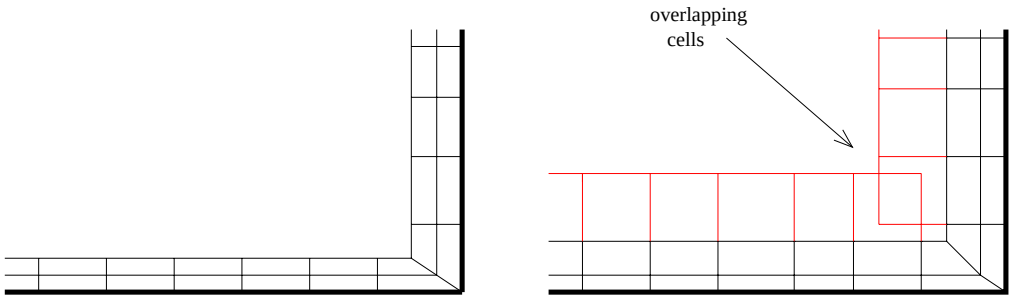


Figure 3.7: Two-dimensional sketch of a boundary layer grid in a concave region

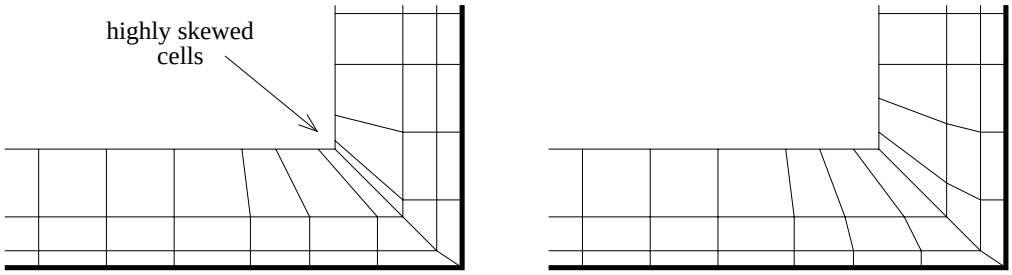


Figure 3.8: Comparison between prismatic grids having a high (left) of low (right) number of normal layers. Note how an excessive number of normal layers might reduce the quality of the exterior layers.

approach is the generation of cells with higher skewness (see Fig. 3.8, left). Suitable smoothing techniques should then be used to locally improve the grid quality (see Fig. 3.8, right). The number of normal layers is an important parameter for a boundary layer grid and can often determine the success for the extrusion of thick prismatic grids on complex geometries. Since the cell skewness is lower for normally grown layers, their number should be as high as possible, but not so high that the quality of the overlying layers is impaired.

As far as convex regions are concerned, the calculation of the local wall normal must be very accurate, in particular when generating thick boundary layer grids. This accuracy is achieved imposing very low tolerances for the geometrical description of wall surfaces, thus increasing the number of facets discretizing each surface. In convex regions where curvature suddenly increases (this is the case of a winglet tip, for example), a too high boundary layer can lead to discontinuous changes in cell dimensions, as illustrated in Fig. 3.9. The grid spacing on the body surface should then be modified in these zones, in order to obtain more uniform outer layers.

In our simulations, the prismatic boundary layer grid has been grown from the bulb, keel and winglets surfaces. The height of the first layer is 0.01 mm. The number of layers in the prismatic grid is 25 and their thickness grows geometrically with a ratio of 1.3. The maximum thickness of boundary-layers was found to be limited by overlapping problems in the region of bulb-keel connection.

In the final step of the grid generation procedure, a Delaunay-type algorithm is used to fill the external volume with tetrahedrons.

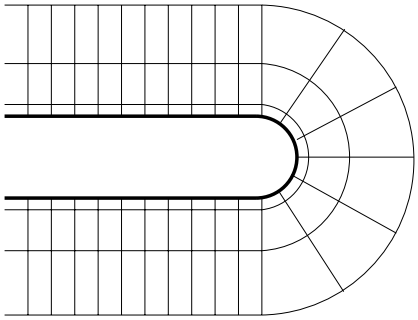


Figure 3.9: Two-dimensional sketch of boundary layer grid on a convex surface.

3.5 Numerical simulations of the flow around yacht appendages

A key factor for the success of an America's Cup yacht is the design of a set of appendages which can guarantee high performances in different sailing configurations (upwind and downwind) over a large range of wind conditions (and, consequently, of boat speeds). On the one hand, the drag should be minimized in order to obtain the highest downwind speed, on the other hand, in upwind sailing, the efficiency (lift/drag) should be maximized. Other factors, such as heeling stability and structural resistance constraints, must be taken into account as well.

A broad variety of different appendage configurations have been analysed. The flow around the hull and a full set of appendages common to IACC racing yachts (including keel, bulb, winglets, and rudder) has been simulated, considering a static flat water surface.

The domain employed in all the simulations has length equal to 200 m, width equal to 150 m and depth equal to 50 m. Inlet velocity and and zero normal stress boundary conditions are imposed respectively on the upstream and downstream face of the domain. On the water surface and on the lateral and bottom surfaces of the domain, symmetry boundary conditions are imposed (see Section 1.1.1 and Fig. 3.10).

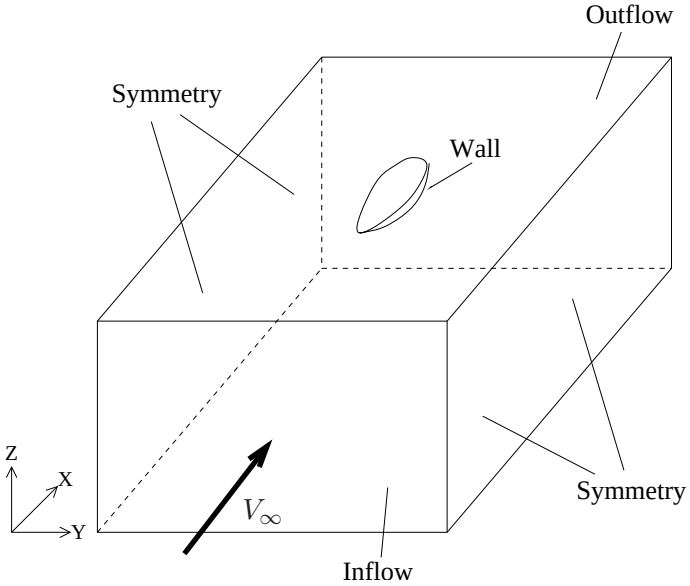


Figure 3.10: Boundary conditions adopted for the simulations of the flow around yacht appendages.

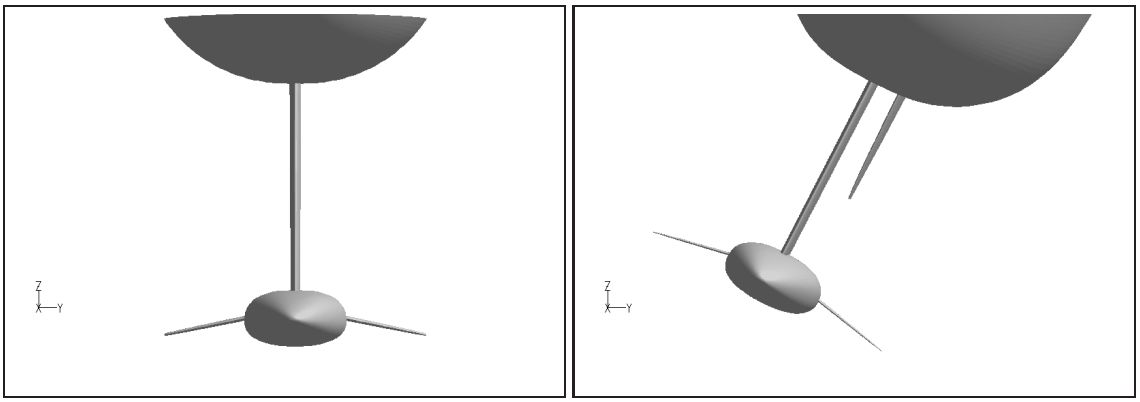


Figure 3.11: Yacht appendices: downwind configuration (left) and upwind configuration (right).

The density and viscosity of the sea water in the Hauraki Gulf, Aukland, host city of the 2003 America's Cup edition, have been used as physical properties of the water in our simulations. These values are $\rho_w = 1025.8$ kg/m^3 and $\mu_w = 0.001081$ kg/ms, respectively.

Most of the computations were performed using sixteen processors of an Origin 3800 parallel machine (see Section 2.4). For the downwind configuration, the total number of grid cells is about 4.5 million, where about 2.5 million cells are contained within the boundary layer to better take into account the viscous effects. For the upwind configuration, where the rudder is included as well, we used grids with about 5.5 million cells, where 3 million cells were contained within the boundary layer. Two examples of downwind and upwind configurations are shown if Fig. 3.11. Computational times to reach steady state varied between 10 and 14 hours per run (see Tab. 3.1).

Configuration	**Downwind**	**Upwind**
Geometry	Hull/Keel/Winglets	Hull/Keel/Winglets/Rudder
Surface grid	180,000 faces	200,000 faces
Volume grid	4,500,000 elements	5,500,000 elements
Runtime	10 hr	14 hr

Table 3.1: Mesh sizes and computational times

3.5.1 Bulbs

The design parameters defining a bulb shape include lateral and vertical profiles as well as the cross section. Advantages and disadvantages of slender (long) versus fuller (short) bulb shapes have been assessed by testing different configurations. Longer bulbs usually perform better with respect to the pressure drag; on the other hand, they have larger wetted surface which increases the viscous component of the drag. The pressure drag D_p and the viscous drag D_v are defined, respectively, by:

$$D_p = \int_{\partial B_w} -p\boldsymbol{n}\cdot\boldsymbol{i}\,dA,$$

$$D_v = \int_{\partial B_w} \big(\mu(\boldsymbol{\nabla u}+\boldsymbol{\nabla u}^T)\cdot\boldsymbol{n}\big)\cdot\boldsymbol{i}\,dA,$$

where ∂B_w is the wetted surface of the boat, $\boldsymbol{i}$ is the unit vector in x direction and $\boldsymbol{n}$ is the unit vector normal to the boat surface and pointing into the computational domain.

The influence of different cross section shapes has also been investigated. In particular, we have analysed the effect of lowering the center of gravity by using cross sections more elliptical than the traditional one, with the long axis running port-starboard. The potential advantage is that a lower center of gravity increases the righting moment of the yacht. Also in this case, the numerical simulations of several bulbs with different cross sections has given useful indications about the trade-off between advantages and disadvantages of each configuration. Other characteristics, such as the bulb camber and the shape of the tail region, have been analysed as well.

Bulb profile

We present the results of a first set of computations that have been performed to compare two bulb configurations with different lateral profiles and cross sections. A fully turbulent flow is considered. The first configuration, referred to as *Bulb1*, is characterised by a higher total length L and a smaller surface frontal area S than the second configuration *Bulb2*. In particular, we consider the following ratios between the geometric characteristics of the two configurations:

$$\frac{L_{Bulb1}}{L_{Bulb2}} = 1.3$$

$$\frac{S_{Bulb1}}{S_{Bulb2}} = 0.8$$

A sketch of the lateral views of the two bulbs considered is shown in Fig. 3.12.

In order to guarantee the correct incoming flow conditions (highly dependent on the presence of other components of the boat), the geometry considered includes one of the two bulbs, the keel and the hull. For each bulb, we have then considered two different configurations:

- a *downwind* configuration: symmetric (yaw angle $\beta_Y = 0°$, heel angle $\beta_H = 0°$)
- an *upwind* configuration: non-symmetric (yaw angle $\beta_Y = 1°$, heel angle $\beta_H = 25°$)

Moreover, the computations have been performed over a range of boat speeds ($V_b = 7, 8, 10, 12$ kts for the *downwind* configuration and $V_b = 8, 10$ kts for the *upwind* configuration).

Tables 3.2 and 3.3 summarize the viscous and pressure components of the drag computed on the keel and the two bulbs for the *downwind* configuration at various boat speeds. As expected, *Bulb1* (the long and slender one) always displays a lower pressure drag and a higher viscous drag (due to its larger wetted surface). Moreover, within the range of speeds considered, Bulb1 seems to guarantee a lower total drag.

Figure 3.12: Lateral sections of the two bulb configurations: *Bulb1* (left) and *Bulb2* (right)

	Keel			*Bulb1*		
Boat Speed [kts]	Pres.	Visc.	Total	Pres.	Visc.	Total
7	70.7	109.7	180.4	64.7	186.1	250.8
8	91.5	140.2	231.7	84.3	239.0	323.3
10	141.0	212.3	353.3	136.0	362.0	497.6
12	201.5	299.0	500.5	187.9	505.9	693.8

Table 3.2: Drag (N) on keel and *Bulb1* downwind configuration at different boat speeds.

	Keel			*Bulb2*		
Boat Speed [kts]	Pres.	Visc.	Total	Pres.	Visc.	Total
7	69.0	107.6	176.6	106.9	173.3	280.1
8	89.2	137.6	226.8	139.2	222.4	361.6
10	137.6	208.4	346.0	221.1	336.0	557.1
12	196.6	293.5	490.1	310.1	470.9	781.0

Table 3.3: Drag (N) on keel and *Bulb2* downwind configuration at different boat speeds.

Tables 3.4 and 3.5 reports the drag components computed in the *upwind* configuration for the two bulbs. The trend observed for the *downwind* configuration is confirmed (*Bulb1* guarantees a lower total drag ($\approx -13\%$)). On the other hand, when *upwind* configurations are considered, the capacity of a set appendages to produce side force (lift) should also be considered, in order to evaluate their efficiency. The lift components for the two bulbs are reported in Table 3.6 and 3.7. We can note that, although *Bulb2* supplies a lift which is 5-6% higher than the one supplied by *Bulb1*, the slenderness of *Bulb1* leads to have a longer keel so that the total lift force for *Bulb1* appendage configuration is 1% higher.

Some useful indications about the considered geometries can also be extracted from visualizations. In particular, observing the streamlines around the nose of the two bulbs on the symmetry plane of the boat (see Fig. 3.13), we can state that both *Bulb1* and *Bulb2* are well aligned with the flow field. The flow reach the nose of both bulbs with a local angle of attack lower than 1 degree with respect to the local coordinate system. Streamline visualizations have also been used to better understand the behaviour of the vortex generated at the tail of the bulb (see Fig. 3.14). The whole matrix of the presented computations indicates that *Bulb1* performs globally better than *Bulb2*.

	Keel			*Bulb1*		
Boat Speed [kts]	Pres.	Visc.	Total	Pres.	Visc.	Total
8	270.2	148.8	419.0	145.6	246.7	392.2
10	421.0	225.4	646.4	227.7	372.8	600.5

Table 3.4: Drag (N) on keel and *Bulb1* upwind configuration at different boat speeds.

	Keel			*Bulb2*		
Boat Speed [kts]	Pres.	Visc.	Total	Pres.	Visc.	Total
8	267.2	145.5	412.7	212.2	227.3	439.4
10	416.3	220.5	636.8	331.4	343.1	674.4

Table 3.5: Drag (N) on keel and *Bulb2* upwind configuration at different boat speeds.

	Keel			*Bulb1*		
Boat Speed [kts]	Pres.	Visc.	Total	Pres.	Visc.	Total
8	9025.9	-2.6	9023.4	951.3	0.9	952.1
10	14147.8	-3.8	14144.0	1490.0	1.3	1491.3

Table 3.6: Lift (N) on keel and *Bulb1* upwind configuration at different boat speeds.

	Keel			*Bulb2*		
Boat Speed [kts]	Pres.	Visc.	Total	Pres.	Visc.	Total
8	8908.1	-2.2	8905.9	1019.8	0.1	1019.9
10	13962.7	-3.3	13959.4	1597.0	0.2	1597.2

Table 3.7: Lift (N) on keel and *Bulb2* upwind configuration at different boat speeds.

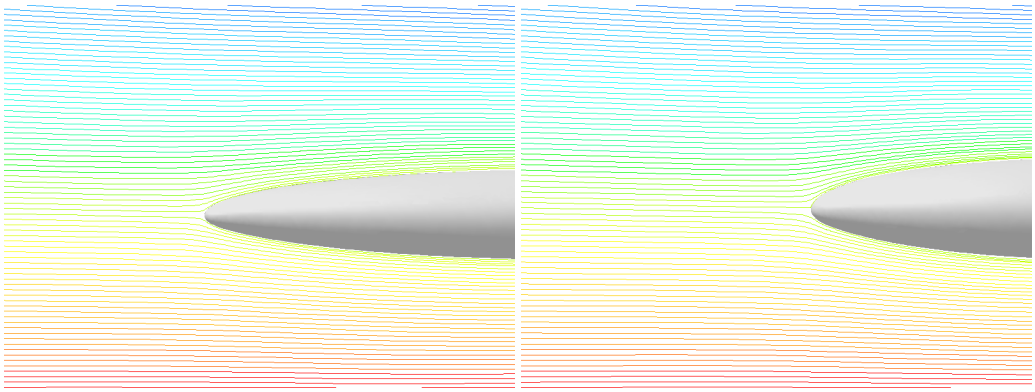


Figure 3.13: Streamlines on the symmetry plane at the nose of *Bulb1* (left) and *Bulb2* (right).

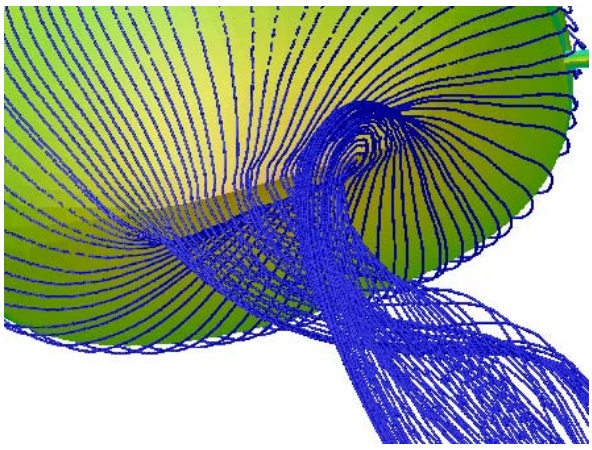


Figure 3.14: Streamlines showing the vortex detaching from the bulb tail.

Laminar-turbulent flow modelling

When compared with experimental predictions on similar body shapes, the results obtained in the bulb simulations discussed above have been found to overestimate the drag. This can be due to the fact that a fully turbulent flow was considered. Unfortunately, as discussed in Section 1.3, reliable laminar-turbulent transition simulations are still hard to be obtained with RANS solvers. To overcome this limitation, a possible approach is to fix laminar zones around the appendages where the turbulent eddy viscosity is set to zero. The extension of such zones has to be previously determined by experimental measurements or by using simplified models (see Section 3.3.2).

We have adopted this strategy to compute the forces acting on the two bulb configurations described above. The extension of the laminar zones around bulb and keel for one of the configuration considered are presented in Fig. 3.15 For the present calculations, the approximate transition locations have been determined using full scale experimental measurements combined with results obtained by the boundary layer code 3C3D^{TM}.

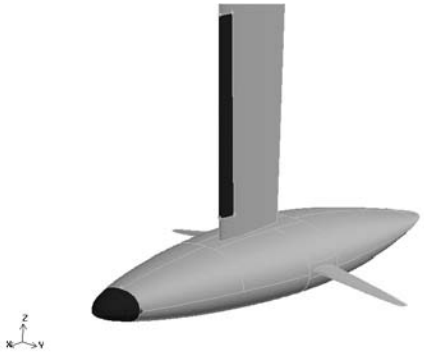


Figure 3.15: Imposed laminar zones (dark) on keel and bulb.

The drag acting on the keel and on the two bulbs in *downwind* configuration with boat speed $V_b = 10$ kts is presented in Tables 3.8 and 3.9. The drag components associated with the laminar-turbulent computation are reported and compared with the corresponding values obtained from the fully turbulent simulation.

The introduction of the laminar-turbulent transition region on the bulb decreases the total drag

		Keel			*Bulb1*	
Flow	Pres.	Visc.	Total	Pres.	Visc.	Total
Fully Turbulent	141.0	212.3	353.3	136.0	362.0	496.6
Laminar-Turbulent	128.2	174.7	302.9	132.0	355.4	487.5

Table 3.8: Drag (N) on keel and *Bulb1* downwind configuration with fully turbulent and laminar-turbulent flows.

		Keel			*Bulb2*	
Flow	Pres.	Visc.	Total	Pres.	Visc.	Total
Fully Turbulent	137.6	208.4	346.0	221.1	336.0	557.1
Laminar-Turbulent	126.7	169.0	295.7	220.1	319.8	539.9

Table 3.9: Drag (N) on keel and *Bulb2* downwind configuration with fully turbulent and laminar-turbulent flows.

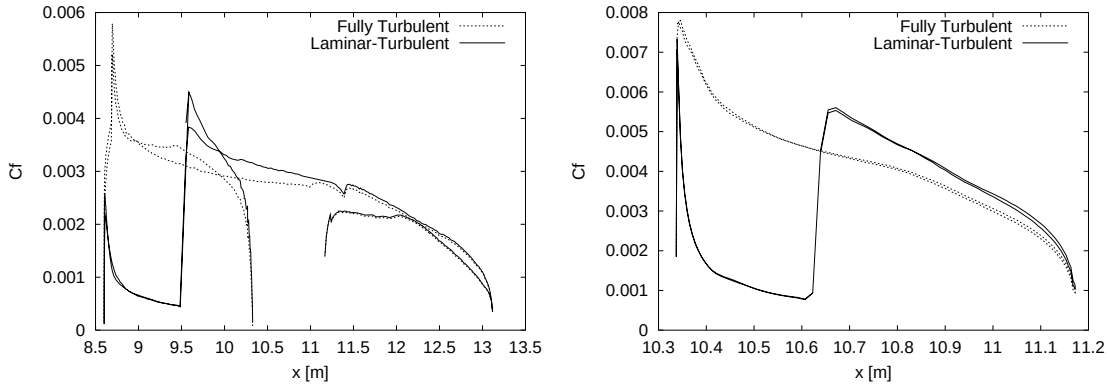


Figure 3.16: Skin friction coefficient on the symmetry plane of the bulb (left) and on the keel section at $z = -3$ m for the full turbulent case and for the laminar-turbulent case.

by around 2% for *Bulb1* (3% for *Bulb2*). This reduction is almost completely due to the viscous component of the drag. The decrease is slightly larger for *Bulb2* since its laminar region is larger. The behaviour of the skin friction coefficient on the symmetry plane of *Bulb1* is displayed in Fig. 3.16 (left), for both fully turbulent and laminar-turbulent cases.

The introduction of a laminar zone has a more remarkable effect on the drag acting on the keel, since its shape has been designed in order to have a high laminarity. The viscous component of the drag on the keel is around 15% smaller than in the fully turbulent case and the pressure drag also decreases (by around 8%). The effect on the pressure drag can be associated with the reduction of the boundary layer thickness.

The skin friction coefficient on the keel (at the $z = -3$ m section) for the fully turbulent case and for the laminar-turbulent case are presented in Fig. 3.16 (right). We can note that the imposition of a laminar zone has a relevant influence on the skin friction distribution, resulting in the decrease of viscous component that is observed in the forse estimates.

Tables 3.10-3.13 present the drag and lift components acting on the appendages in *upwind* configurations for *Bulb1* and *Bulb2* when fully turbulent and laminar-turbulent flows are considered.

The variation of the drag component of the forces due to the modelling of the laminar-turbulent

	Keel			*Bulb1*		
Flow	Pres.	Visc.	Total	Pres.	Visc.	Total
Fully Turbulent	421.0	225.4	646.4	227.7	372.8	600.5
Laminar-Turbulent	415.3	198.3	613.6	219.8	359.6	579.4

Table 3.10: Drag (N) on keel and *Bulb1* upwind configuration with fully turbulent and laminar-turbulent flows.

	Keel			*Bulb2*		
Flow	Pres.	Visc.	Total	Pres.	Visc.	Total
Fully Turbulent	416.3	220.5	636.8	331.4	343.1	674.4
Laminar-Turbulent	413.6	193.1	606.8	320.1	326.1	646.6

Table 3.11: Drag (N) on keel and *Bulb2* upwind configuration with fully turbulent and laminar-turbulent flows.

	Keel			*Bulb1*		
Flow	Pres.	Visc.	Total	Pres.	Visc.	Total
Fully Turbulent	14147.8	-3.8	14144.0	1490.0	1.3	1491.3
Laminar-Turbulent	14069.2	1.0	14070.2	1459.5	0.8	1460.3

Table 3.12: Lift (N) on keel and *Bulb1* upwind configuration with fully turbulent and laminar-turbulent flows.

	Keel			*Bulb2*		
Flow	Pres.	Visc.	Total	Pres.	Visc.	Total
Fully Turbulent	13962.7	-3.3	13959.4	1597.0	0.2	1597.2
Laminar-Turbulent	13792.2	1.2	13793.5	1570.4	-1.3	1569.1

Table 3.13: Lift (N) on keel and *Bulb2* upwind configuration with fully turbulent and laminar-turbulent flows.

transition in *upwind* configuration, are very similar (in term of percentage) to that observed for the *downwind* case. The effect of the transition model on the lift component of the force is not very important neither on the keel (less that 1%) nor on the bulb (1% on both *Bulb1* and *Bulb2*).

In summary, it has been observed that modelling the laminar-turbulent transition can be worthwhile in order to obtain more reliable results, when streamlined geometries with relevant region of laminarity are considered.

3.5.2 Winglets

Since 1983, when they were first adopted by the America's Cup winner Australia II, winged-keels have been extensively employed in IACC yacht design. Similarly as in aircraft applications (see Fig. 3.17), the underlying idea is that the presence of winglets at the tip of a keel (or an aircraft wing) can reduce the lift-induced drag.

Figure 3.17: Learjet Model 28/29, first production jet aircraft to utilize winglets (Photo from NASA Archive).

Lift-induced drag is a consequence of producing lift by a finite wing (in our case, the keel). The lift production is given by the pressure differential between the windward side and the leeward side of the keel. This pressure differential induces a flow around the bulb from the high-pressure

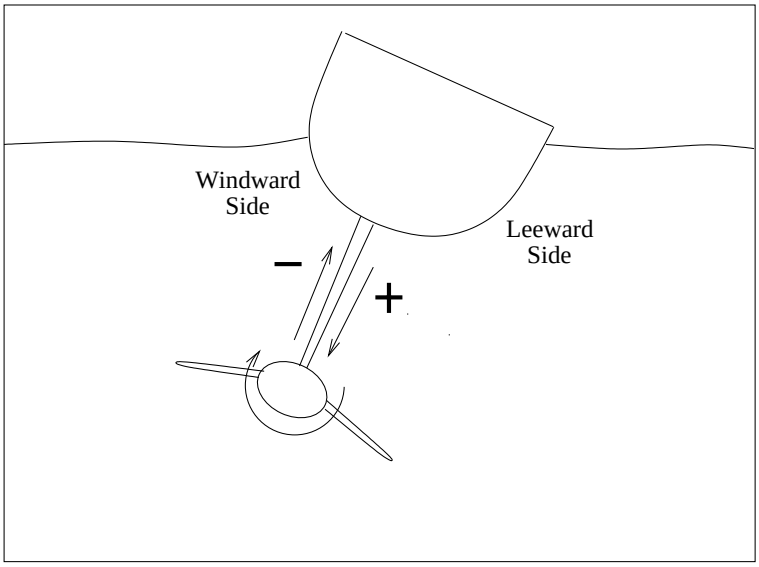


Figure 3.18: Spanwise flow and tip vortex induced by the pressure differential across the keel in upwind sailing.

water on the leeward side of the keel to the low-pressure water on the windward side (see Fig. 3.18). The flow leaving the leeward side moves down while that on the windward surface moves up. When they meet all along the trailing edge of the keel, they generate a vortical motion that, within a short distance downstream, is concentrated into the tip vortex detaching from the bulb tail (see Fig. 3.19). Clearly, the generation of this vortex requires energy. The transfer of this energy from the appendages to the water is the lift-induced drag.

The idea underlying the use of winglets (both in aerodynamic and sailing applications) is to produce

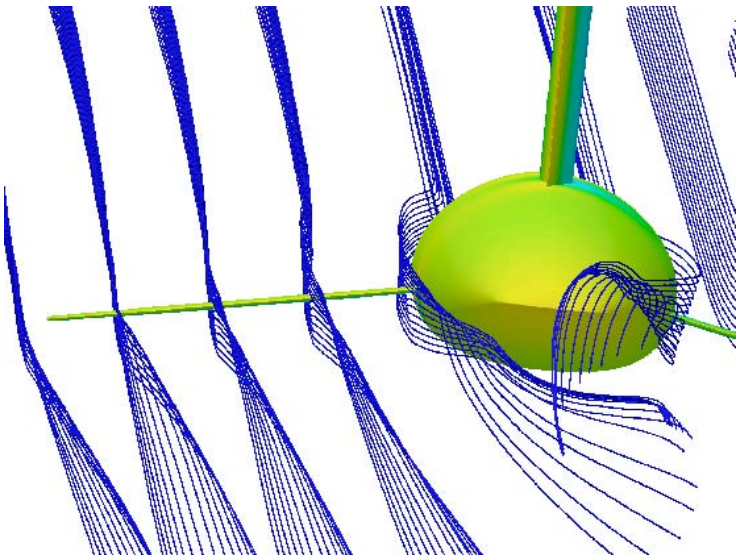


Figure 3.19: Surface pressure contours on the yacht appendages and streamlines around the winglets.

a flowfield that interacts with that of the keel to reduce the amount of spanwise flow. In other words, the spanwise velocities induced by the winglets oppose, and therefore partially cancel, those generated by the keel. The winglet spreads out the influence of the bulb tail vortex such that the induced drag is reduced.

The lift-induced can be expressed as follows:

$$C_{D_i} = \frac{C_L^2}{\pi \Lambda_{\text{eff}}} \tag{3.3}$$

where C_L is the lift coefficient and Λ_{eff} is the effective keel aspect ratio. For a longer keel (with a higher aspect ratio), the tip effect on the inboard portions of the keel is reduced and, consequently, the lift-induced drag is reduced. Unfortunatly, the length of the keel is limited by the America's Cup rules. The adoption of winglets induces an increase in the effective aspect ratio Λ_{eff}, thus reducing the lift-induced drag component on the appendage system.

Moreover, in a sailing boat, winglets can also provide some thrust in the boat course direction when the boat pitches and heaves in waves. In fact, their lift vectors are rotated forward by an angle equivalent to the local induced flow field, resulting in a forward thrust on the winglets (see [Mil98]). In spite of their extensive use in the last 20 years, an optimal winglet design still remains an open problem in the yacht design community.

Several winglet design parameters have been considered in the present work: longitudinal position, angle of attack and sweep angles. The information that can be extracted from these simulations are both quantitative (forces, pressure coefficient distributions, wall stress distributions) and qualitative (vector fields and streamlines visualizations to locate, e.g., regions with flow separations).

Longitudinal winglet position

The first winglet parameter that we have consider is their longitudinal location along the bulb symmetry plane (front versus aft location). This analysis was expected to supply a better understanding of the difference between the two configurations considered. The forces acting on the whole set of appendages have been computed and their efficiency in reducing the vorticity which escapes from the back of the bulb has been analysed. A sketch of the two winglet configurations can be seen in Fig. 3.20.

For both winglet configurations (aft and front), we have considered, as in the previous section, a *downwind* and a *upwind* configuration. Laminar regions are imposed on bulb and keel, while the flow on the winglets is assumed to be fully turbulent.

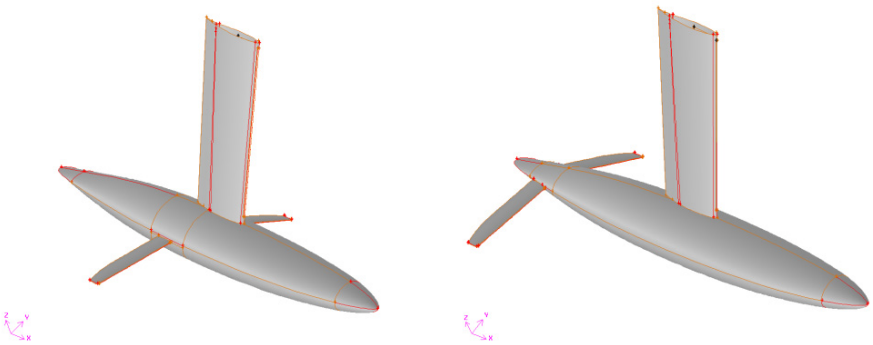


Figure 3.20: Front winglets (left) and aft winglets (right) configurations.

	Aft Winglets		*Front Winglets*	
Component	Drag	Lift	Drag	Lift
Keel	305.6	-55.2	306.9	-55.0
Bulb	445.0	178.0	449.3	326.8
Leeward Wing	51.1	323.1	59.8	320.9
Windward Wing	50.8	324.0	59.7	320.9

Table 3.14: Drag and lift (N) on the different appendage components for aft and front winglets configurations with boat speed $V_b = 10$ kts in *downwind* configuration.

The drag and the (vertical) lift components of the forces acting on each appendage element, for both aft and front winglets configurations are presented in Tab. 3.14. These values refer to the *downwind* case with boat velocity $V_B = 10$ kts.

We can observe that, for both the aft and front positions, positive and non-negligible values of the lift force are obtained (about 320 N). This is clearly non optimal since, as discussed above, the winglets should have a beneficial effect in *upwind* sailing without penalizing too much the performances *downwind*. Such high values of lift are responsible of a relevant induced component of the drag. A better behaviour in this configuration can be achieved by reducing the mean value of the angle of incidence. No major differences between the two locations (which in the present study has the same geometry) are apparent when analysing the *downwind* configuration.

Drag, (lateral) side force and (vertical) lift on each appendage element in the *upwind* configuration are reported in Tab. 3.15.

The trade-off between the advantages expected from the front winglets (higher capability of improving the fin keel effectiveness by e.g. increasing the effective aspect ratio) and those expected from the aft winglets (better performance in term of the forces acting on themselves, due to the stronger incidence angles that these winglets are subject to) can be clearly observed in the set of results presented in Tables 3.14 and 3.15. We can point out that the aft position seems to guarantee an advantage in term of drag. In fact, while the values of total drag on keel are comparable for the two configurations, the drag on the winglets is found to be different, with a non-negligible advantage for the aft position, which can be associated with the higher incidence due to a stronger rolling-up process.

Moreover, as expected, the front winglets lead to a stronger increase of the side force on the keel, and therefore to a higher effective aspect ratio of the keel itself. However, when the side force given by the winglets is also considered, we can see that the side force gained by the keel with the winglets in front position (around 120 N) is balanced by the larger side force on the winglets in aft position.

	Aft Winglets			*Front Winglets*		
Component	Drag	Side	Lift	Drag	Side	Lift
Keel	604.0	14033.2	7208.3	607.0	14158.6	7596.9
Bulb	516.5	1542.2	862.7	516.4	1559.6	1014.0
Rudder	310.3	2525.7	1323.1	314.2	2494.9	1367.1
Leeward Wing	34.6	498.0	-341.8	53.8	347.5	-467.7
Windward Wing	10.7	-228.7	1021.7	44.9	-194.2	717.5

Table 3.15: Drag, side force and lift (N) on the different appendage components for aft and front winglets configurations with boat speed $V_b = 10$ kts *upwind* configuration.

For the aft position in upwind configuration, the flow presented a separation bubble on the bottom (suction) side of the leeward winglet. The separated region is displayed in Figs 3.21 and 3.22. The slight advantage of the aft position configuration that seems to emerge from the results presented here may be further improved if this flow separation (which clearly reduces the effectiveness of the lifting surface) is avoided.

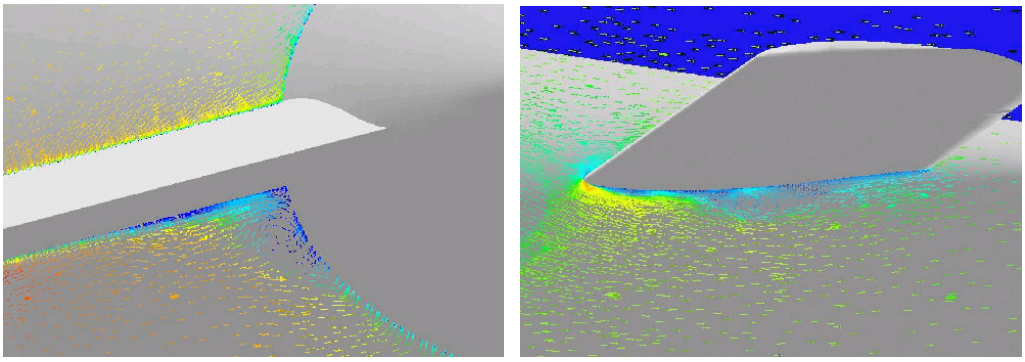


Figure 3.21: Vector field in the region of separated flow.

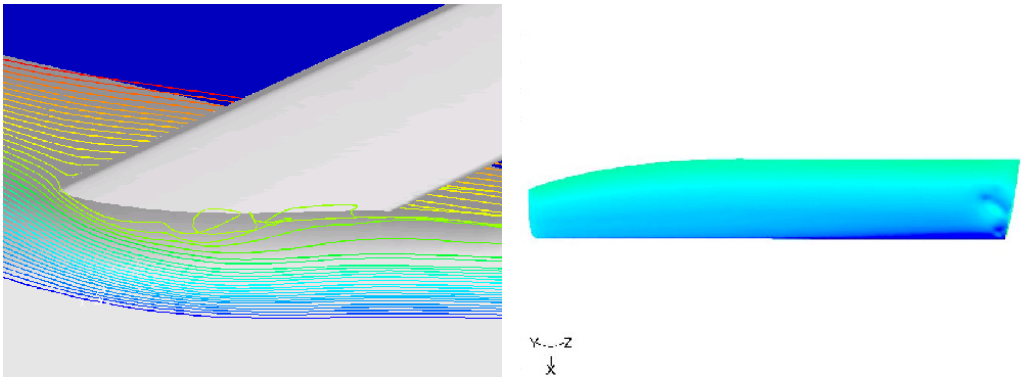


Figure 3.22: Streamlines (left) and pressure distribution (right) in the region of separated flow.

Information can be also extracted from the analysis of the wake evolution behind the appendages. In particular, we analysed the maps of longitudinal vorticity at different longitudinal distances in the wake of the appendages. Some of these maps are reported in Fig 3.23. These images show that the aft winglet configuration seems to be able to spread the vorticity more efficiently along the whole winglet span reducing the concentration of vorticity escaping from the bulb tail. Such behaviour is usually associated to a lower induced drag. In both cases, the vorticity distributed along the winglet span gets entrained by the stronger bulb trailing vortex. It can be noted that this process is delayed in the aft position case, when compared to the front position case. The same trend can be deduced observing the evolution of the total pressure in the wake of the appendages (see Fig. 3.24). Indeed, the lost of total pressure is related to the energy dissipation due to the vorticity effect in the wake.

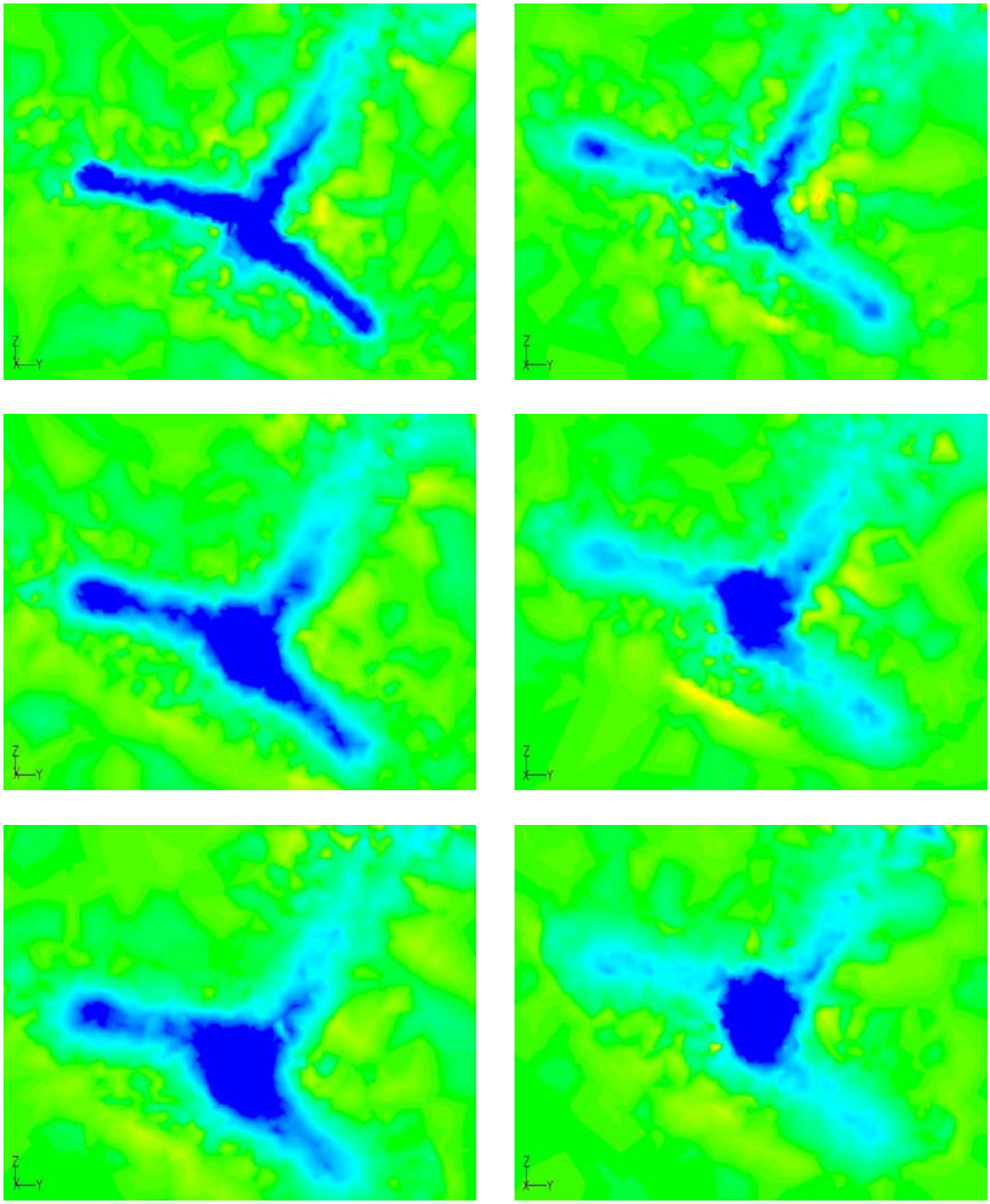


Figure 3.23: Distribution of the x component of vorticity on different longitudinal cut planes in the wake of the appendages: aft position (left column) and front position (right column).

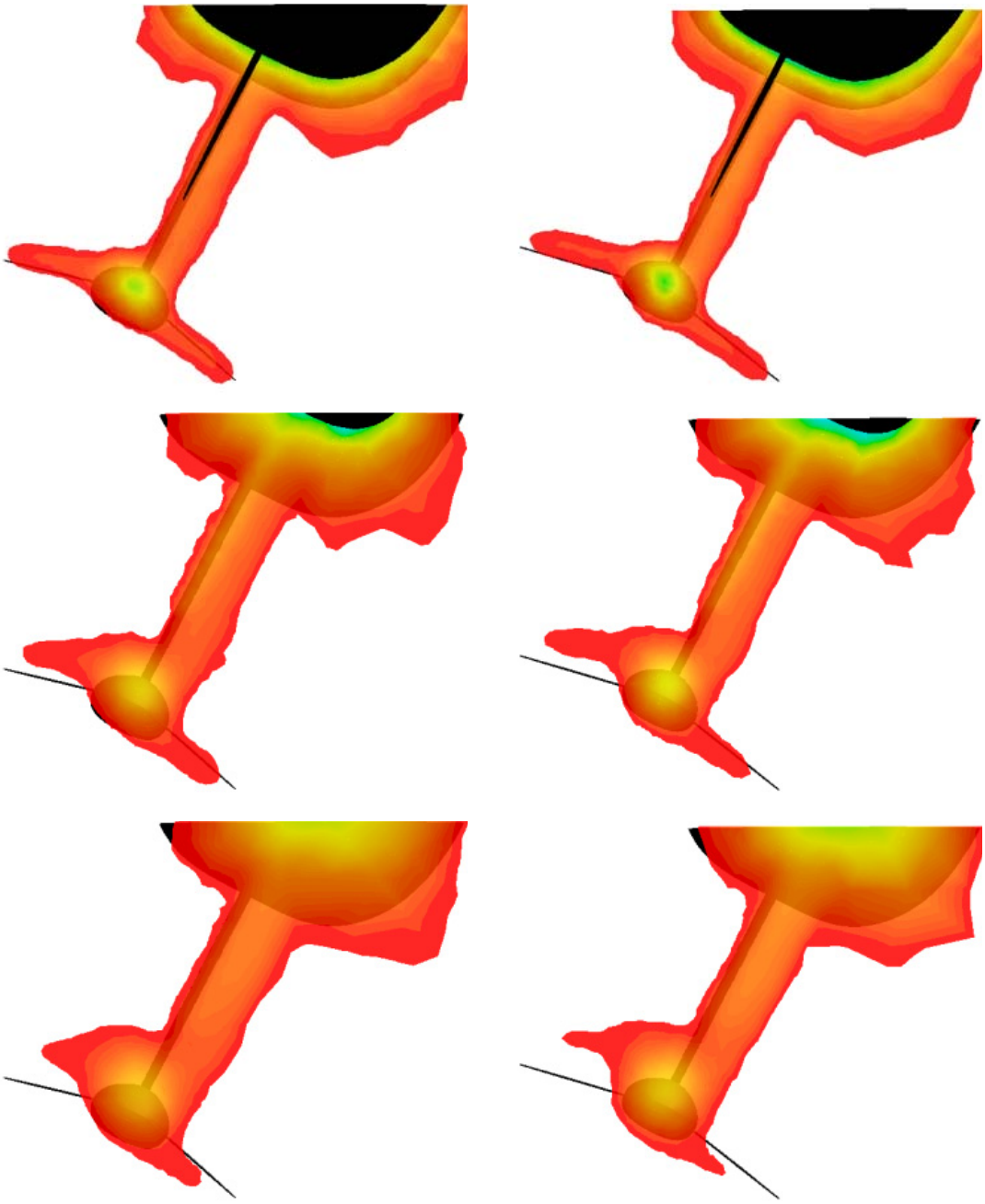


Figure 3.24: Total pressure distribution on different longitudinal cut planes in the wake of the appendages: aft position (left column) and front position (right column).

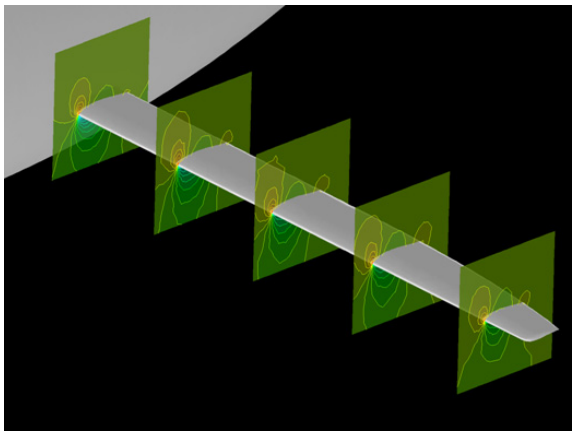


Figure 3.25: Pressure distribution on 5 cut planes across the winglet.

From the post-processing of the results, several different information, useful for the design process, can be extracted. In the specific case of the winglets, for example, visualizations as the one presented in Fig. 3.25, showing the pressure distribution on several cut planes across the winglet, can be usefull to extract information about the local orientation of the flow. In particular, the distribution of local angle of attack along the winglet span can help design a non-uniform distribution of geometrical angle of attack (twisting) and a non-uniform distribution of section shape along the span.

Geometrical angle of attack

The vertical force produced by the winglets in the *downwind* configuration has been found to produce excessive induced drag. To reduce this effect, a range of winglet angle of attack positions has been considered. The objective is to investigate the overall benefit on round-the-buoys performance of reducing the vertical loads in the *downwind* sailing for both front and aft winglet positions.

In Fig. 3.26, the load distributions along the two winglet positions in *downwind* configuration are presented. The span distributions of local lift coefficient for three different geometrical angles of attack are displayed. The local lift is the force component normal to the planform. Analogous results for the *upwind* configuration are presented in Fig. 3.27.

An evaluation of the efficiency of a specific design choice is not straightforward when *downwind* and *upwind* performances are considered singularly. One possible approach is based on the definition of a global performance index based on both *downwind* and *upwind* performances suitably weighted. We have considered an index E based on *downwind* drag and on *upwind* appendage efficiency (side force/drag):

$$E = \frac{\alpha}{D_D} + \beta \frac{S_U}{D_U},$$

where D_D is the *downwind* drag (expressed in Newton), S_U and D_U are, respectively, the *upwind* side force and drag (N). The two weighting parameters α and β depend on the effective time spent in the two sailing condition during a standard match race. These coefficients have been defined based on the performances of previous design configurations and on the Design Team experience. The values considered in the present study are $\alpha = 4000$ N and $\beta = 3/5$.

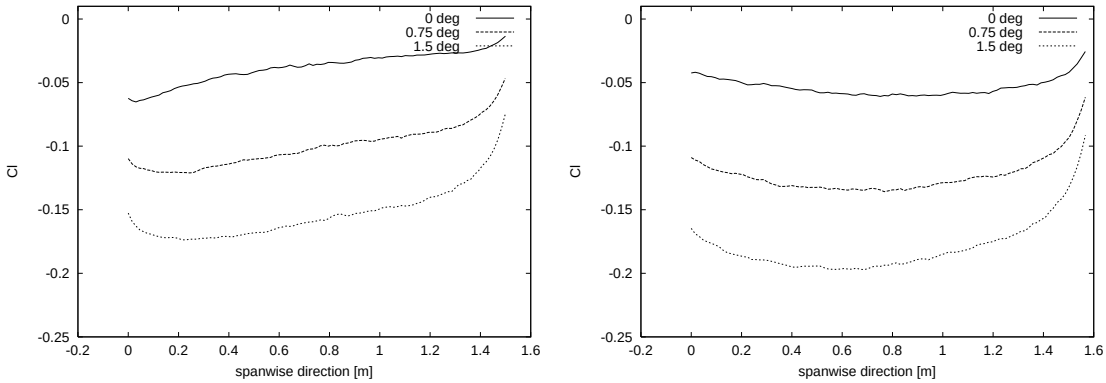


Figure 3.26: Spanwise distribution of the local lift coefficient for the aft (left) and front (right) winglet position in downwind condition.

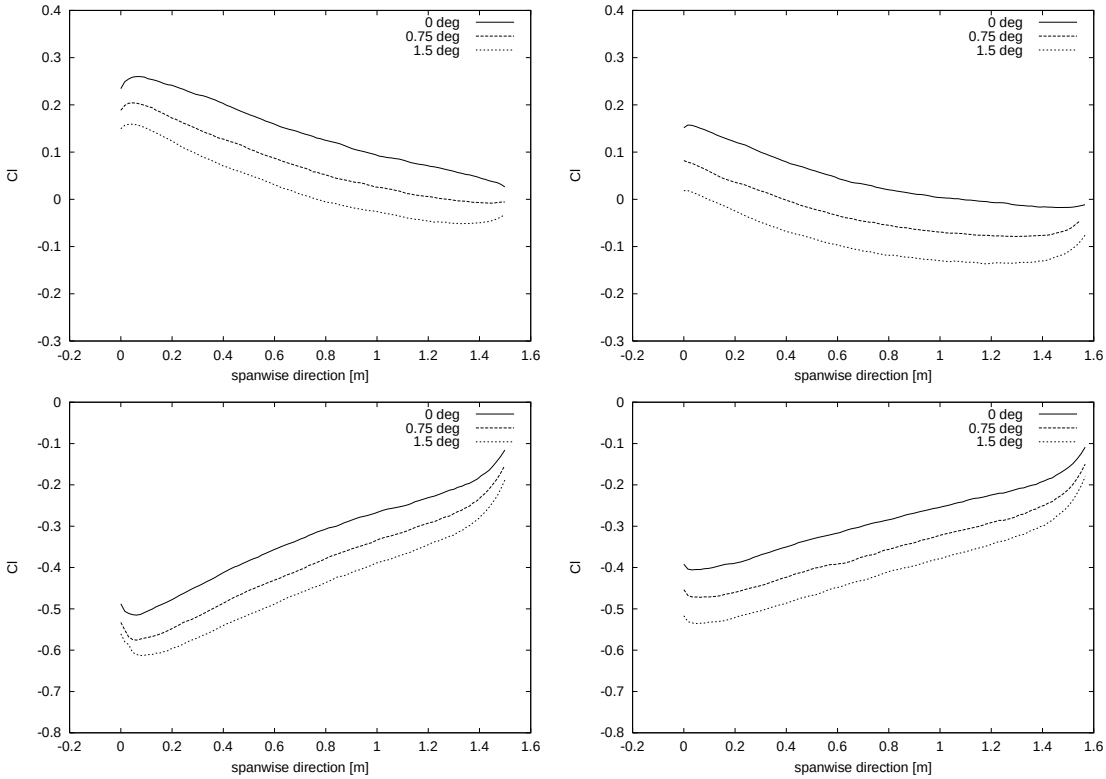


Figure 3.27: Spanwise distribution of the local lift coefficient for the aft (left) and front (right) winglet position in upwind condition. The top row refers to the windward winglet and the bottom row to the leeward winglet.

Analysing the global performance index E for the three geometrical angles of attack considered ($\theta = 0°, 0.75°$ and $1.5°$), it has been found that a local optimum occurs within the range of angles considered (see Table 3.16). In particular, in both aft and front configurations, the maximum of the global performance index E is attained for $\theta = 0.75°$.

Configuration	$\theta[°]$	D_D	D_U	S_U	E
Aft Winglets	0	882.1	1100.1	19789.3	15.32
	0.75	884.7	1114.4	20247.7	15.42
	1.5	901.8	1126.4	20477.6	15.34
Front Winglets	0	906.3	1136.8	20064.2	15.00
	0.75	909.4	1145.6	20414.9	15.09
	1.5	912.6	1158.4	20666.8	15.08

Table 3.16: Drag (N), side force and performace index E for aft and front winglet configurations with different geometrical angles of attack.

Winglet sweep angle

The winglets are placed in the vortex of the keel/bulb system and thus they are subject to a large spanwise incidence angle. The results obtained in the angle of attack study showed that the load distributions on the winglets were far from elliptical. This was not unexpected since the spanwise variation of both the incident angle of attack and velocity magnitude is quite large. In order to recover an optimal loading profile it is necessary to either twist the winglet or alter the planform characteristics (by modifying the chord distribution or the sweep angle). In fact, a reduction of the winglet induced drag can be reached moving the loading toward the tips. It is likely that twist would have a larger adverse effect in the downwind condition since the spanwise flow variation in this case is greatly decreased.

A preliminary investigation on the influence of the sweep angle on the global performances has been performed. We have considered two aft winglets configurations with sweep angle $\phi = 0°$ and $\phi = 15°$, respectively. The results in terms of drag, side force and global performance index are presented in Tab. 3.17. We can note a beneficial effect when the sweep angle is increased. Further investigations would be necessary to achieve an optimal configuration with respect to this parameter.

$\phi[°]$	D_D	D_U	S_U	E
0	884.7	1114.4	20247.7	15.42
15	882.3	1102.6	20120.1	15.48

Table 3.17: Drag (N), side force and performance index E for aft winglets configurations with different sweep angles.

3.6 Numerical simulations of the flow around yacht sails

The mathematical model and numerical method described in Chapter 1 and 2, as well as the grid generation strategies discussed above, are also suitable for the numerical simulation of the aerodynamic flow around the sails. In fact, for the wind and boat velocities considered, the flow can still be assumed as incompressible. Moreover, the typical Reynolds number based on the characteristic length of the sail ($L = 10$ m), on the true wind speed ($V_{TW} = 20$kts ≈ 10 m/s) and on the air physical properties, can be estimated as:

$$\mathrm{Re} = \frac{\rho_a \, L \, V_{TW}}{\mu_a} \approx \mathcal{O}(10^6),$$

corresponding to a turbulent flow. The same turbulence models considered for the hydrodynamic simulations can be adopted.

As in the hydrodynamic studies, numerical simulations have been used to evaluate different sail design concepts. We present here the main results concerning a new design for the head of the main sail. In particular, we have analysed the effect of *squaring* the main sail head. The standard *rounded* shape main sail and the *squared* configuration (see Fig. 3.28) have been compared.

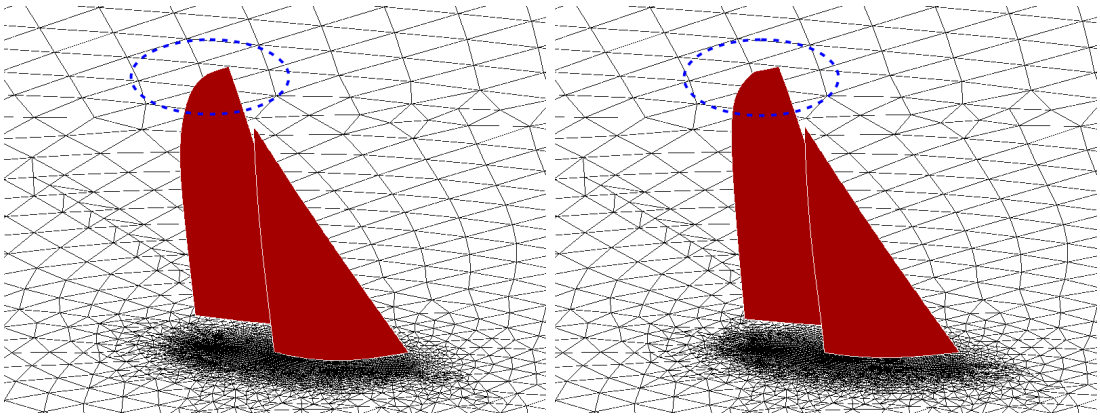


Figure 3.28: Original rounded sail configuration (left) and square head configuration (right).

The inflow velocity boundary conditions take into account the presence of the atmospheric boundary layer. Because the hull was stationary in the simulation, the variation in both the magnitude of the apparent wind and the apparent wind angle was prescribed. This was derived from the boat speed, the sailing angle, and a relation for the true wind speed obtained from an approximation of the atmospheric boundary layer. For this study, the so-called 1/10 law was used to model the true wind speed V_{TW} in this layer,

$$V_{TW}(z) = V_{TW}(10) \left(\frac{z}{10}\right)^{\frac{1}{10}} \tag{3.4}$$

where z is the vertical coordinate in meters measured from the static waterplane and the true wind speed outside the atmospheric boundary layer $V_{TW}(10) = 12$ kts.

The mast bend is different for the different main sails and also the genoa geometry is slightly different. Therefore, a bare numerical comparison of the forces (reported in Tab. 3.18) for the two

	Thrust		Side Force	
	Square Head	Original	Square Head	Original
Main Sail	1184	1164	10005	9953
Genoa	3155	3160	14186	14215
Total	4339	4324	24191	24168

Table 3.18: Comparison [N] of the forces on genoa and main sail computed on the original configuration and on the square head configuration

configurations may not be significant. The results presented in Tab. 3.18 indicate a slightly higher thrust supplied by the square headed main sail with a negligible increase of the side force.

This slightly improved behaviour can be explained analysing the pressure distributions on the pressure and suction sides of the two configurations (see Figs. 3.29 and 3.32)

Observing the pressure distribution on the head of the main sail, it is evident (see detail in Fig. 3.30) that the reshaping of the head increases the region on the pressure side where the positive pressure peak acts. It can also be noted, on the other hand, that the suction effect at the top of the main sail is higher for the original configuration (see 3.32), apparently due to the different trimming of the genoa. However, the force estimates seem to indicates a favorable trade-off for the squared head configuration.

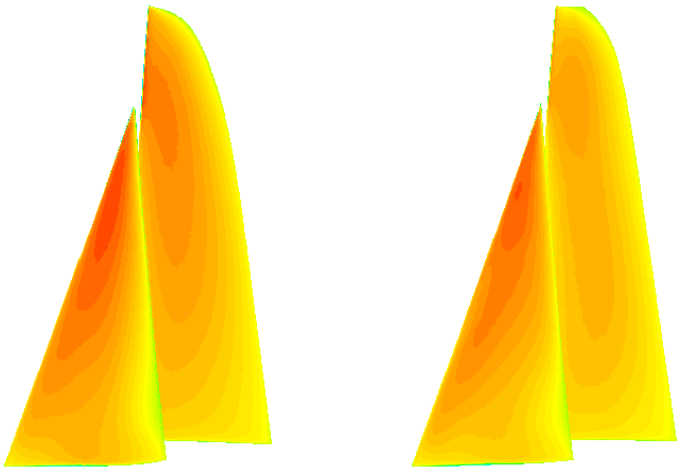


Figure 3.29: Pressure contour on the pressure side: original configuration (left) and square head configuration (right).

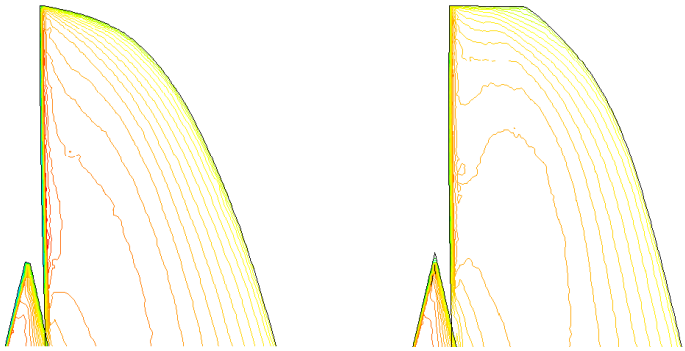


Figure 3.30: Pressure contour on the pressure side (detail of the main sail head): original configuration (left) and square head configuration (right).

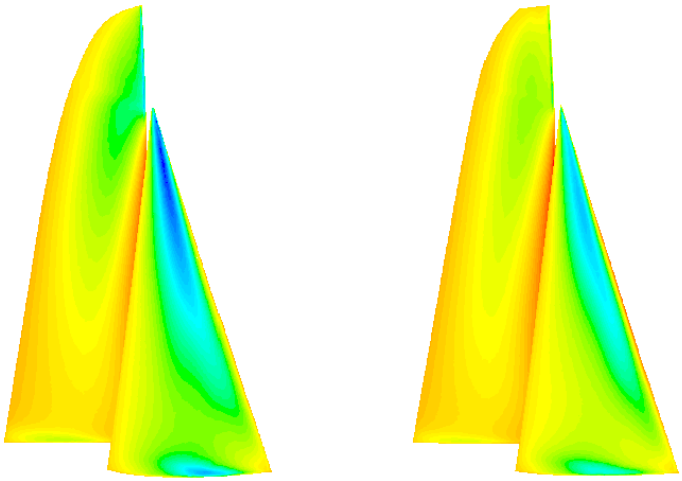


Figure 3.31: Pressure contour on the suction side: original configuration (left) and square head configuration (right).

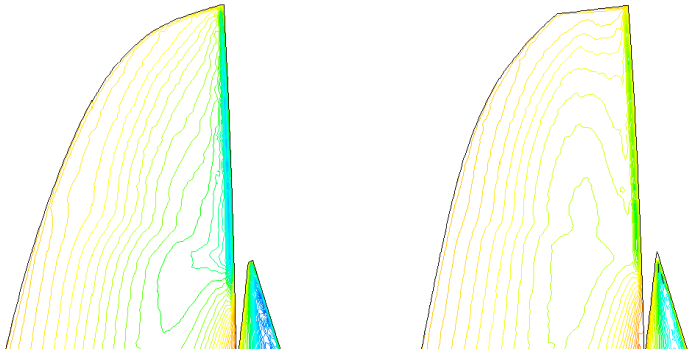


Figure 3.32: Pressure contour on the suction side (detail of the main sail head): original configuration (left) and square head configuration (right).

3.7 Scalability results of parallel computations

In order to assess the parallel performance of the solver, we have performed a study on the parallel scalability, *i.e.* the dependence of the CPU time on the number of processors used for the simulation. The scalability of parallel computations depends on the ratio between computing and communication time. To highlight the different behavior of the parallel performances when grids of different sizes are considered, we have measured the CPU time required to run a simulation using different numbers of processors on three grids composed of 1400000 cells, 2500000 and 3850000 elements, respectively. The same simulation has been run on a number of processors varying between 1 and 16 on each grid. The partitioning of the computational domain has been performed using the code Metis. Two examples of partitioning in 4 and 8 sub-domains for the finer grid are displayed in Fig. 3.33.

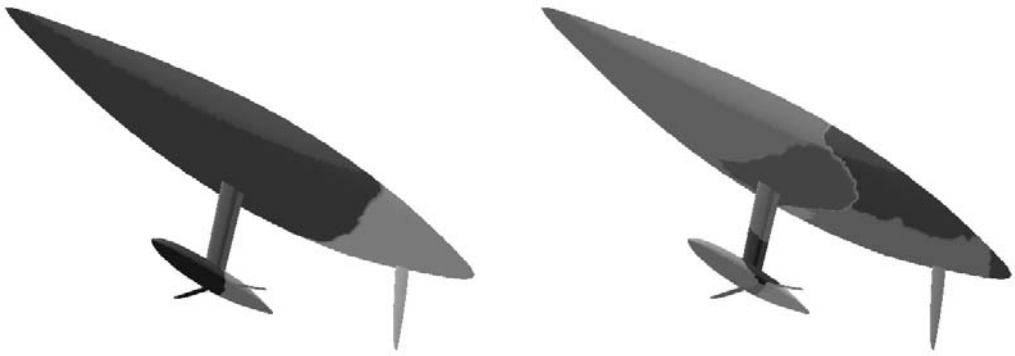


Figure 3.33: Partitioning of the domain into 4 (left) and 8 (right) sub-domains For the sake of clarity, only the surface is shown.

The CPU times versus number of processors are presented in Fig. 3.34, Fig. 3.35 and Fig. 3.36 for simulations on the small size, medium size, and large size grid respectively.

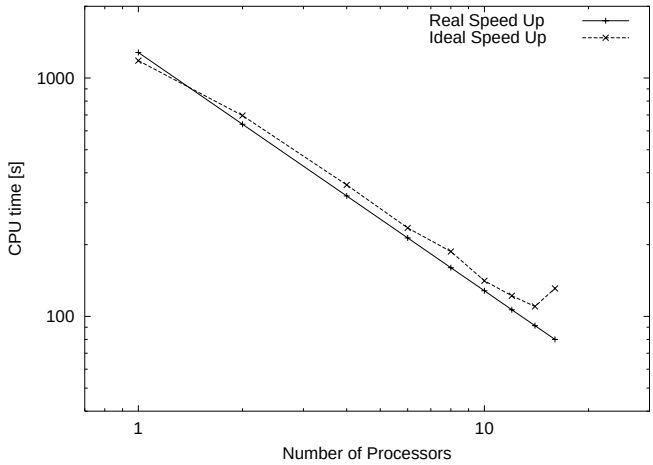


Figure 3.34: CPU time versus number of processors (small grid).

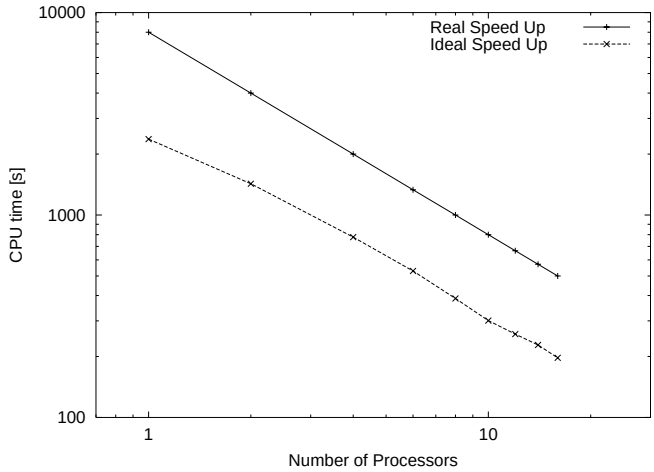


Figure 3.35: CPU time versus number of processors (medium grid).

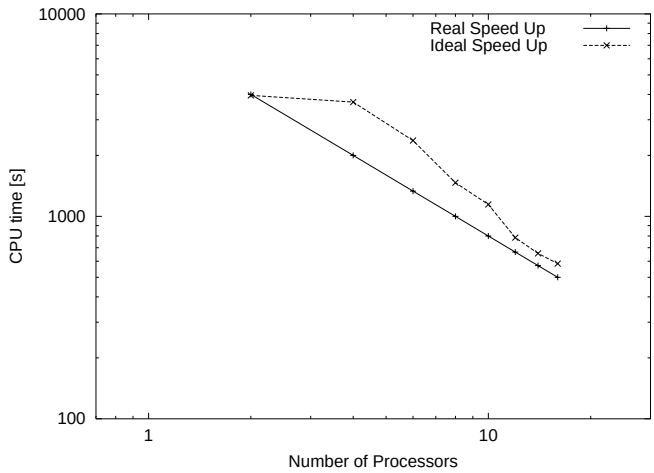


Figure 3.36: CPU time needed versus number of processors (large grid).

As expected, these results show that, on the one hand, when solving ”small” size problems, the use of a large number of processors become useless due to the increasing role of the communication time (Fig. 3.34). On the other hand, for large scale computations the speed up becomes optimal only if a sufficient number of processors is used (Fig. 3.36).

Chapter 4

Numerical Simulation of Free Surface Viscous Flows

The numerical simulations presented in Chapter 3 have been performed neglecting the effect of the free-surface. Although one may speculate that this assumption can be acceptable when the subject of investigation is the flow around sails and appendages of a sailing yacht, this is no longer the case when the focus is on hull performances. The component of the pressure drag associated to the deformation of the free-surface around the hull represents an important contribution to the overall resistance. This drag component is associated to the energy that the hull consumes to maintain the wave field.

The earliest mathematical model for the description of ship waves dates back to Lord Kelvin's theory at the end of nineteenth century [Kel86]. In this work, the wave pattern generated by a point of pressure moving across the water surface was considered. From this pressure perturbation, two separate systems of waves are generated: a divergent wave system emanating from the point with an angle of $19°28'$ and a transverse system which follows the pressure point (see Fig. 4.1). Indeed, these structure can be recognized looking at the wave pattern of most floating bodied in motion.

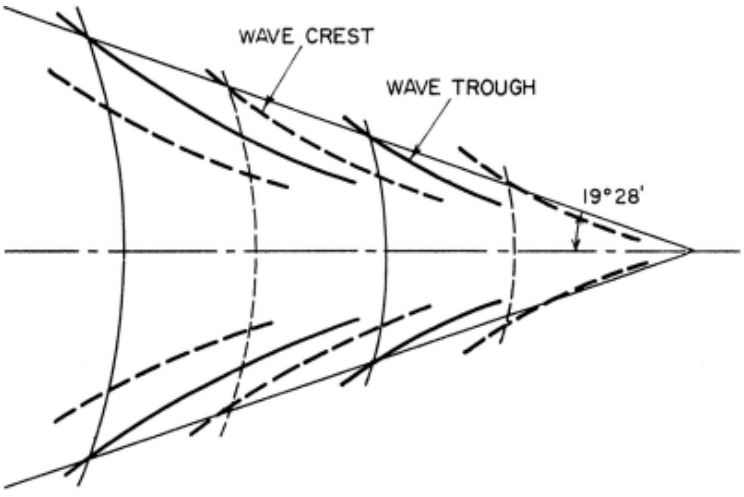


Figure 4.1: Sketch of the Kelvin wave system.

Denoting with V_b the boat speed, the wavelength of the transverse waves is

$$\lambda = \frac{2\pi V_b^2}{g}, \tag{4.1}$$

which corresponds to the wavelength of a deep water wave moving with speed V_b.

The wave field of a boat can qualitatively be described using Kelvin's theory. Approximating the boat by two pressure points separated by a distance L (equal to the ship length), it is possible to derive the wavelength of the transverse wave system generated by the two points, as follows:

$$\frac{\lambda}{L} = 2\pi\,\mathrm{Fr}^2, \tag{4.2}$$

where Fr$= V_b/\sqrt{gL}$ is the Froude number. The wavelength of the resulting wave train only depend on the Froude number.

The Froude number defines the ratio between inertial and gravitational forces. This non-dimensional quantity plays a crucial role in ship hydrodynamics. For example, experimental tests in water tanks has to be run using the same Froude number as in the full scale in order to reproduce the free-surface dynamics. Indeed, one of the main drawbacks in towing tests relies on the difficulty to match simultaneously the Reynolds number and the Froude number, unless a full scale model is used. This is one of the reasons of the increasing role that Computational Fluid Dynamics is assuming in the naval engineering community, as discussed in Section 1.2.

In this chapter, we present numerical simulations of viscous free-surface flows for two test cases, from which we can obtain a first validation of the numerical scheme. Further numerical results on free-surface naval problems will be discussed in Chapter 5.

4.1 A three-dimensional free-surface test case

A first assessment of the numerical method is performed on a three dimensional test case, consisting in the solution of the free-surface flow around a vertical hydrofoil piercing the water-air interface. The hydrofoil has a NACA 0024 profile. The test case setup reproduces the experiment carried out at the IIHR (Iowa Institute of Hydraulic Research) towing tank and reported in [MLS01]. A sketch of the computational domain is shown in Fig. 4.2, together with the dimensions in the three directions.

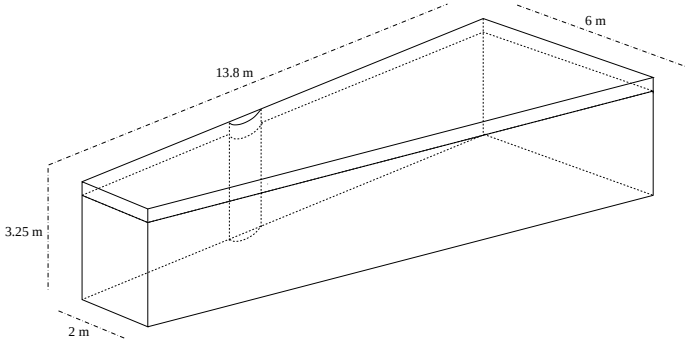


Figure 4.2: Sketch of the computational domain for the piercing body test case.

	Cells	X	Y	Z
Grid 1	13455	40	16	24
Grid 2	115103	80	32	48
Grid 3	951615	160	64	96

Table 4.1: Sizes of the 3 grids for the piercing body test case.

The computational domain is prismatic with a trapezoidal basis (larger at the outflow), in order to reduce problems related to the reflection of waves at the lateral boundary. Note that, due to the symmetry of the problem, only one half of the geometry is considered. For this test case, we have considered an incoming velocity of 1.15 m/s, corresponding to Fr$= 0.37$ and Re$=1.05 \times 10^6$, and air and water in standard condition. The realizable $\kappa - \epsilon$ turbulence method has been used. On the hydrofoil no-slip wall condition is imposed. Inlet Dirichlet boundary conditions are imposed both on the upwind size of the trapezoidal domain and on the lateral inclined side. A symmetry condition is imposed on the symmetry plane and on top and bottom sides. A zero normal stress boundary condition is imposed on the outflow boundary.

Numerical simulations have been performed in order to investigate the dependence of the solution on the grid refinement. Given the relatively simple geometry, a single block structured grid has been used, with two regions of cell clustering located around the interface (to get an appropriate resolution of the free-surface) and around the body (to resolve the boundary layer region). Three grids with different resolutions have been considered. The total number of cells and the number of points along the three directions for different grids are presented in Table 4.1. The mesh is clustered in the vicinity of the static water plane with a minimum vertical mesh spacing of 2 mm for the finer grid. The lateral and top views of the middle size grid are presented in Fig. 4.3.

The wave patterns generated by the hydrofoil with the three different grid resolutions are shown in Fig. 4.4, where it can be noted the dumping effect on the small scale waves using too coarse grids. For the finer grid, a Kelvin's type wave pattern can be recognized.

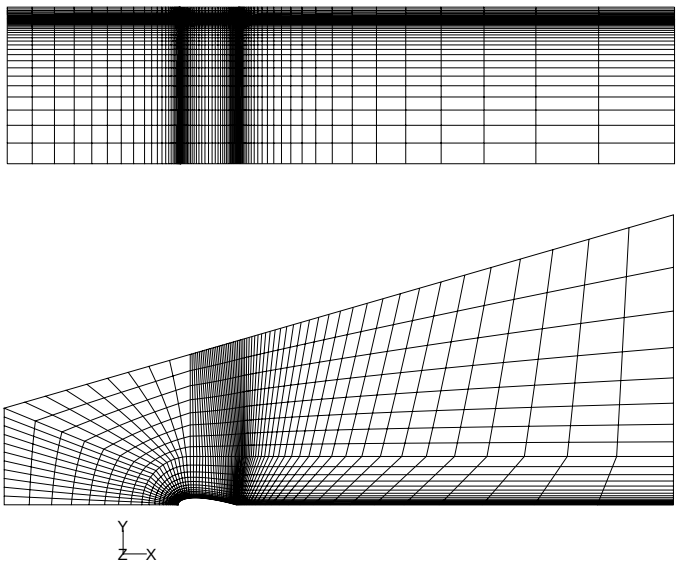


Figure 4.3: Lateral (higher) and top (lower) views of the middle size grid for the piercing body test case.

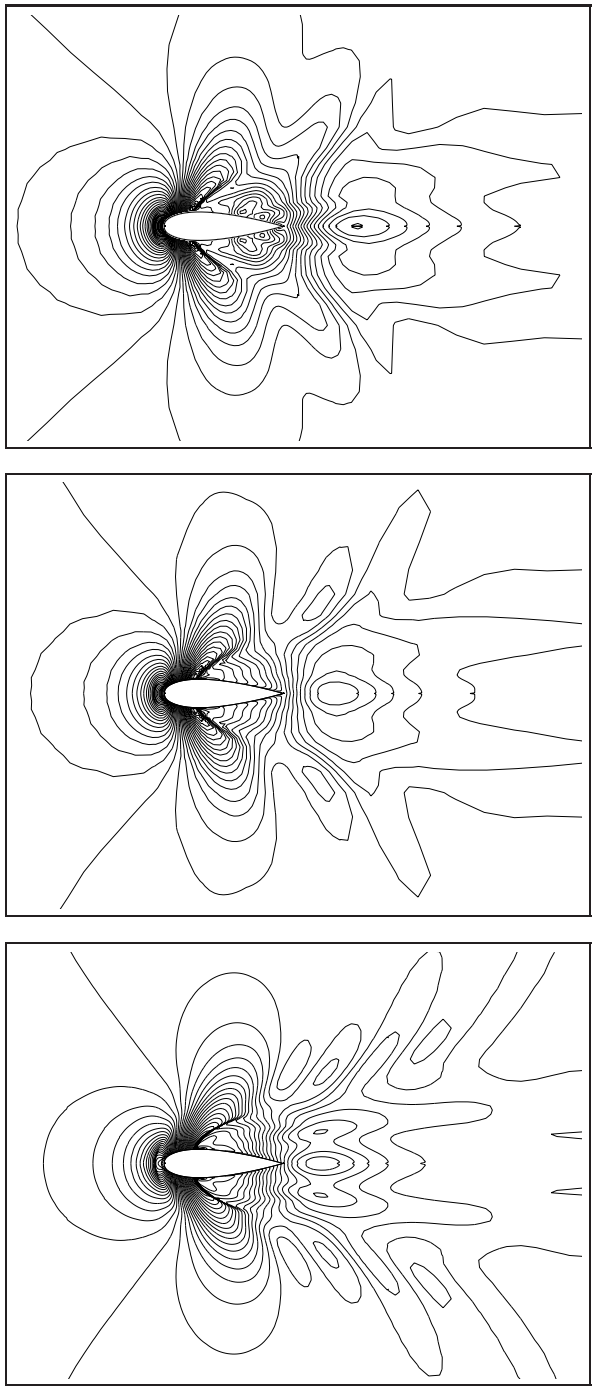


Figure 4.4: Comparison between the wave patterns obtained with the three grids: iso-line of free-surface elevation for (from top to bottom) Grid 1 (coarse), Grid 2 (medium), Grid 3 (fine)

For a quantitative comparison between numerical prediction and experimental results, we consider the wave profile on the hydrofoil surface. In Fig. 4.5, the wave profiles corresponding to the three grids together with the experimental data are displayed. These results show the convergence of the numerical results to the measurements as grid is refined.

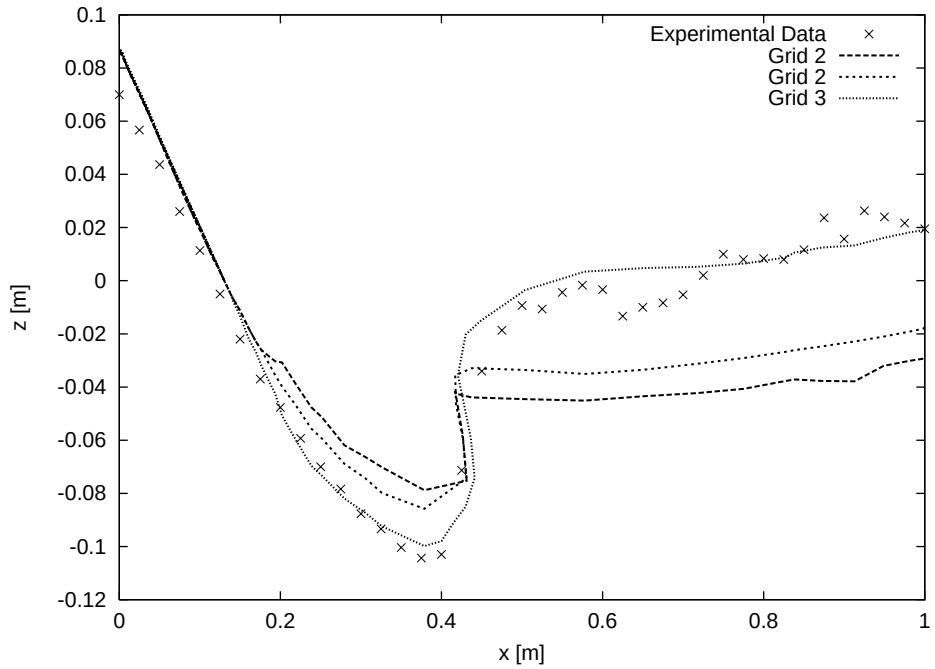


Figure 4.5: Comparison between the numerical and experimental wave profiles on the hydrofoil surface.

As already pointed out in Section 1.2.2, one of the major drawbacks in the Volume of Fluid method is the lack of control on the thickness of the interface, unless suitable (but often computationally too expensive) interface reconstruction algorithms are adopted. In the present case, we can note that the interface is diffused on many grid elements (see Fig. 4.6, right). This effect is reduced, as expected, when lower Froude number, and therefore smaller free-surface deformations, are considered (see Fig. 4.6, left, for a case with Fr=0.2).

Starting from the free-surface solution around the piercing hydrofoil, we have set a simple test case with the aim of showing the influence that an inaccurate prediction of the free-surface location may have on the drag estimation.

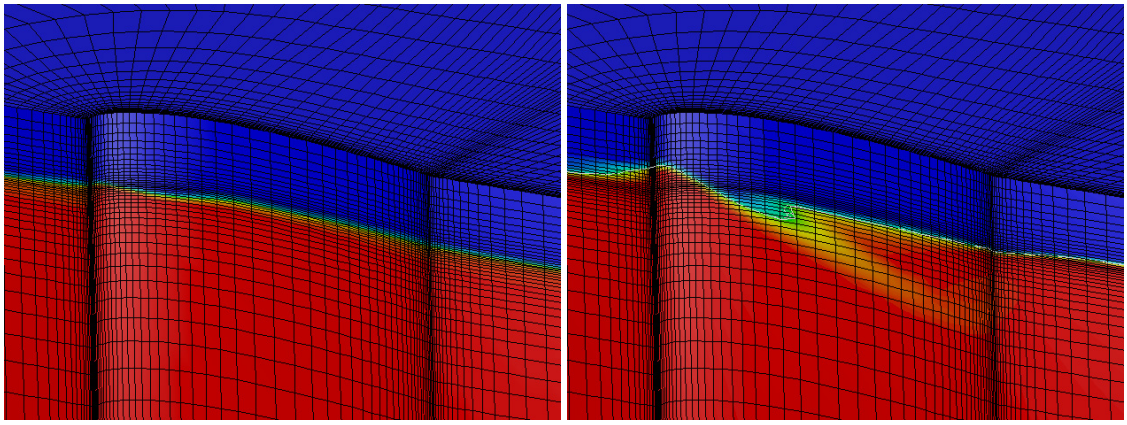


Figure 4.6: Density distributions on the piercing hydrofoil: Fr$=0.2$ (left) and Fr$=0.37$ (right).

We consider the wave pattern around the piercing hydrofoil obtained on the middle size grid for $Fr = 0.2$. The wave amplitude is artificially modified and the dependence of the drag on the wave amplitude is analysed. Five different computational domains have been constructed having as top surface such wave pattern amplified by a factor β (with $\beta = 0, 1, 2, 4, 6$). A one-fluid (water) computation as been performed on the five configurations, with a free-slip boundary condition imposed on the top surface. For $\beta = 0$, the water surface is flat and corresponds to neglecting the free-surface effects. For $\beta = 1$, the water surface is fixed to the wave pattern obtained in the free-surface VOF simulation and the resulting drag estimation is very close to the one obtained in the free-surface simulation.

The velocity magnitude distributions corresponding to different choices of the amplification factor β are displayed in Fig. 4.7.

In Fig. 4.8 the values of pressure, viscous and total drag coefficients as a function of β are presented. The total drag coefficient is defined as

$$C_D = \frac{D}{1/2\rho_w V_\infty^2 A_w}, \tag{4.3}$$

where D is the total drag, ρ is the density of water, V_∞ is the asymptotic velocity and A_w is the wetted area.

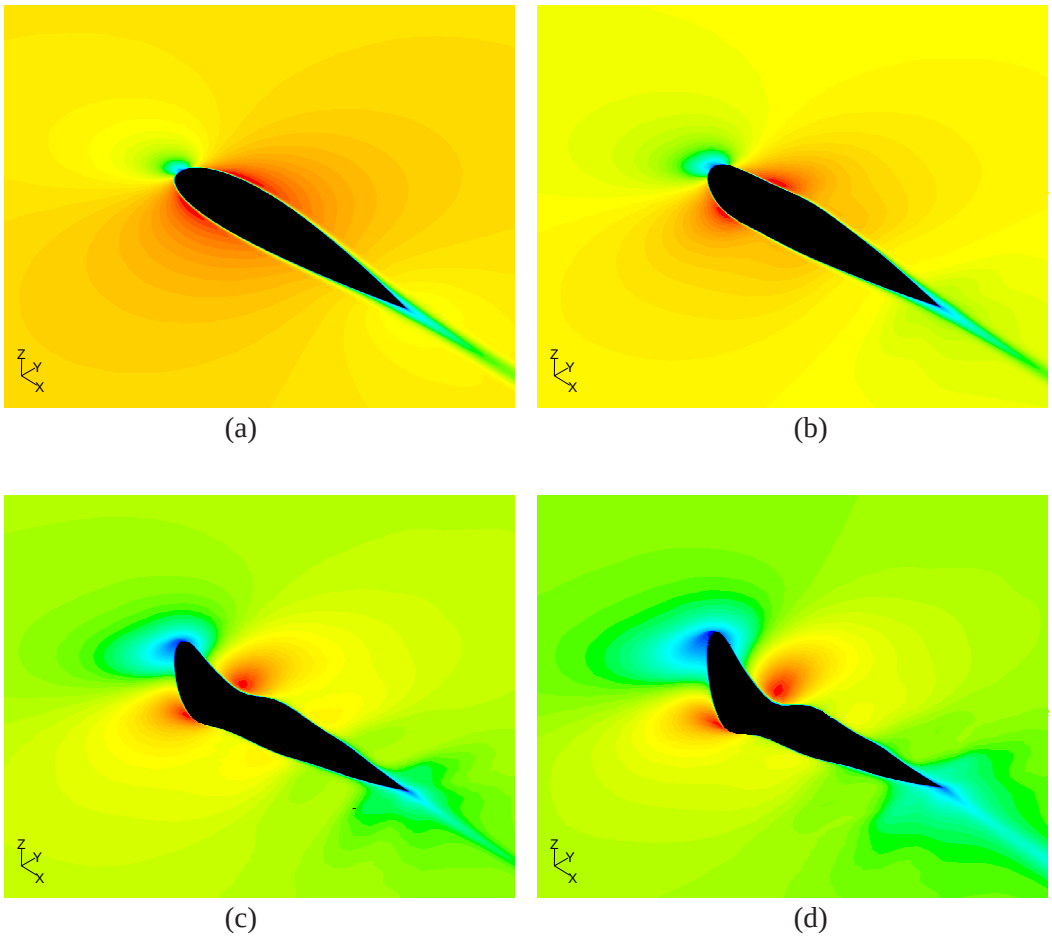


Figure 4.7: Velocity magnitude distributions on the wave surface for different values of the amplification factor β: (a) $\beta = 0$ (flat surface); (b) $\beta = 2$; (c) $\beta = 4$; (d) $\beta = 6$.

These results shows that, as expected, the pressure drag is highly dependent on the wave amplitude, while the viscous component is less sensitive. Even for the low Froude number considered here (Fr=0.2), the differences in pressure drag between cases with $\beta = 0$ and $\beta = 1$ indicates that neglecting the free-surface effects leads to an considerable underestimation of the pressure component (here of the order of 10%).

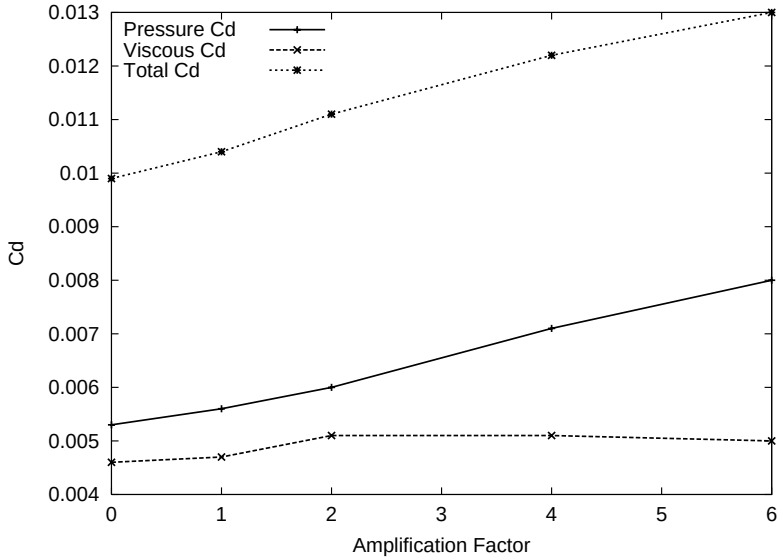


Figure 4.8: Pressure, viscous and total drag coefficients on the piercing hydrofoil as a function of wave amplitude β.

4.2 Wigley hull test case

A classical test case which is often considered in the naval engineering community is the simulation of the free-surface flow around the so-called *Wigley hull* (see, textite.g., [LYOnI99, Azc01, Cow01]). The reason of its popularity is due to its simple geometrical form, which minimizes the efforts for the grid generation, and to the large amount of experimental data available for the validation.

The Wigley hull forms are described analytically by a quadratic equation. The transverse (beam) coordinate is expressed as a function of the longitudinal (x) and vertical (z) coordinates

$$y = \frac{b}{2}\left(1 - \left(\frac{2x}{L}\right)^2\right)\left(1 - \left(\frac{z}{D}\right)^2\right),$$

where b is the beam, D is the draft and L is the hull length. For the present study, the hull specification parameters were chosen to be $b/L = 1/10$, $D/L = 1/16$.

We have considered the experimental data presented in the University of Tokyo contribution to the 1983 Wigley Cooperative Study [KMI+83]. In these experiments, resistance data as well as waterline profiles were obtained using a 2.5 m long model over a large range of Froude numbers (0.1 - 0.408).

We consider a parallelepipedal computational domain situated at -3.75 m < x < 6.25 m, 0 m < y < 5 m, -2 m < z < 0.3 m (see Fig. 4.9). The hull is centered at the origin and is 2.5 m in length. Due

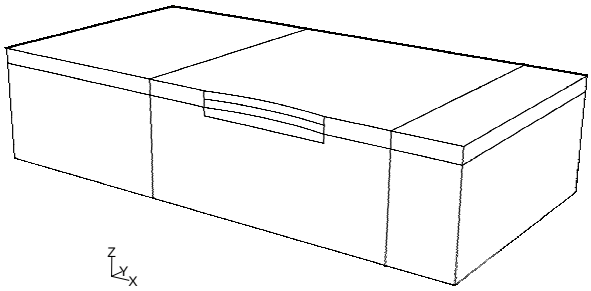


Figure 4.9: Computational domain for the Wigley hull test case.

to the symmetry of the problem with respect to the mid plane, only one half of the geometry is considered. Since both the air and water regions are modelled, the Wigley hull has been extended vertically upward 0.3 m to give it a rather large false free board. The vertical extent of the air zone should be large enough so that the presence of the upper "far-field" boundary does not perturb the resulting solution. Standard boundary conditions (no-slip, inflow, outflow and symmetry) are considered (see Sec. 1.1.1).

Two single-block structured grids with different resolutions have been considered. The total number of cells and the number of points along the three directions for the two grids are reported in Table 4.2. A view of the finer grid is presented in Fig. 4.10.

Two zones of mesh clustering are can be seen in Fig. 4.10: the high-density regions in the boundary layer around the hull surface and the one around the initial position of the interface at the static water plane. The mesh is clustered in the vicinity of the static water plane with a minimum vertical mesh spacing of 3 mm for the finer grid.

Two simulations have been performed on each grid, corresponding respectively to Froude number

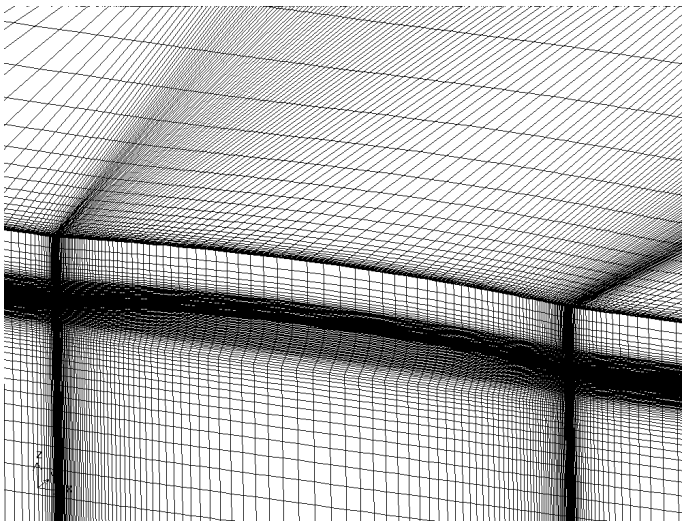


Figure 4.10: Computational grid for the Wigley hull test case.

	Cells	X	Y	Z
Grid 1	720000	151	41	121
Grid 2	1500000	201	51	151

Table 4.2: Dimensions of the 2 grids for the Wigley hull test case.

$Fr = 0.25$ and $Fr = 0.408$. The wave patterns obtained on the finer grid with the two Froude numbers are presented in Fig. 4.11. We can observe the dependence of the wavelength of the wave train on the Froude number, as predicted by Kelvin's theory.

The numerical results have been compared with the experimental data available in [KMI+83]. In particular, in Fig. 4.12 we present the waterline profiles computed on the two grids for Fr=0.25 and Fr=0.408, as well as the corresponding experimental measurements.

For both Froude numbers, the bow wave amplitude appears to be well captured at least with the finer grid. The diffusion of the interface (see Fig. 4.13) leads to a loss of accuracy in the wave profile downstream along the hull, confirming the observations done in the previous section concerning the smearing of the interface.

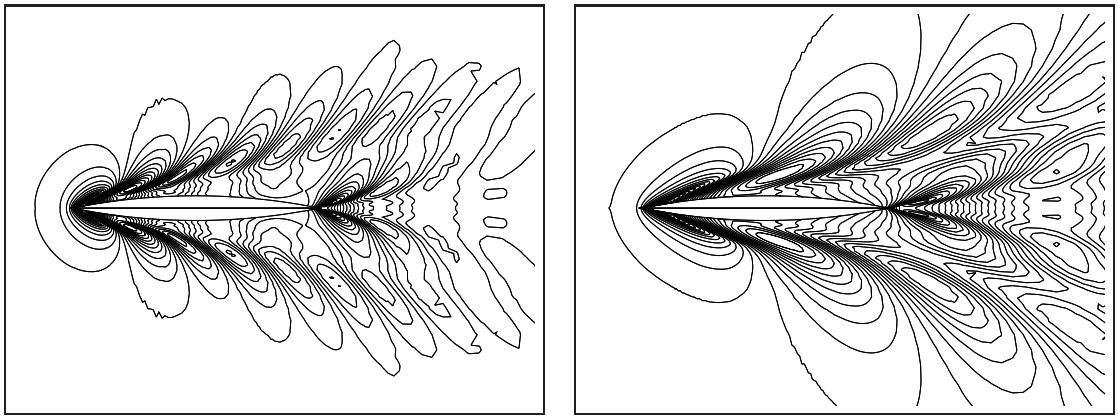


Figure 4.11: Free-surface elevations for the Wigley hull test case: Fr=0.25 (left) and Fr=0.408 (right).

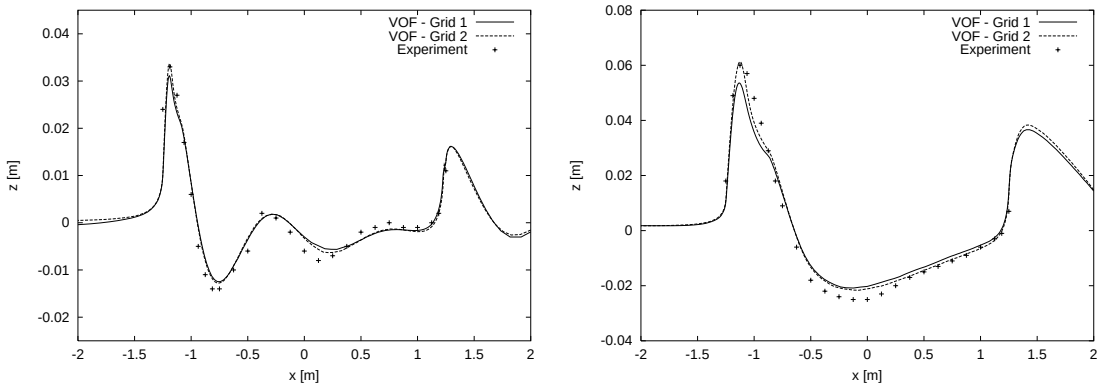


Figure 4.12: Wave profile comparisons for the Wigley hull test case: $Fr = 0.25$ (left), $Fr = 0.408$ (right).

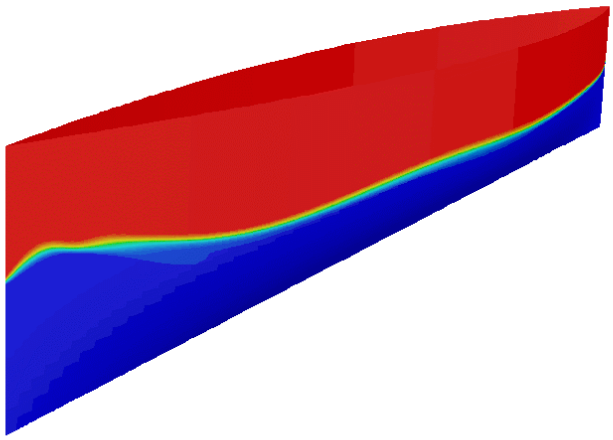


Figure 4.13: Volume fraction distribution on the Wigley hull showing dissipation of the VOF method on the interface (Fr=0.408): air (red) and water (blue).

The values for the total drag coefficient and the relative error with respect to the experimental results are presented in Tab. 4.3.

The relative error for the coarser grid is less than 6% for both Froude number and decrease to less than 1% for the finer grid. Even if these results seem to indicate an acceptable capability of the numerical approach in estimating the total drag, we would like to remark that further improvement could probably be obtained reducing the smearing of the interface region.

	Experiment	Grid 1		Grid 2	
Fr	C_D	C_D	Error (%)	C_D	Error (%)
0.250	4.612	4.431	3.9	4.593	0.4
0.408	5.719	5.522	3.4	5.748	0.5

Table 4.3: Comparison of the numerical and experimental drag coefficients (x 1000) for the Wigley hull test case.

Chapter 5

Numerical Simulation on Olympic Class Rowing Boats

5.1 Introduction

Since centuries, rowing boats have represented an important mean of transport for many human communities. The first illustration of a rowing boat was found in Finland and dates back to 5800 *BC*. As a sport, rowing has old and deep traditions. During the thirteenth century, rowing regattas between gondoliers were held in Venice, Italy. The rowing sport, in the form we know today, was already well assessed in nineteenth century and the famous "Oxford versus Cambridge" university boat race was held for the first time in 1829 on the River Thames. The Olympic program has included men's rowing since 1896 in Athens, but rough sea conditions in the Piraeus harbor forced the events to be canceled at the first Olympic Games and the first Olympic rowing races were run in 1900.

On a rowing boat, the propulsion is guaranteed by the energy that the rowers transfer to the water through the oars. Each oar operates as a lever with the fulcrum located on a fixed point (oarlock) attached to the hull. The stroke cycle can be divided into 4 phases (see Fig 5.1):

- the **catch** is the moment when the oar blade enters the water;
- the **drive phase** is the half of the stroke in which the oar blade is in the water and the rowers move backward applying effort on the oars to move the boat;
- the **finish** is the moment when the oar blade leaves the water;
- the **recovery** is the half of the stroke in which the rower moves for wards and the oars are repositioned for a new stroke.

To maximize the energy supplied during the stroke, the rowers move along the boat longitudinal axis, sliding on moving seats and exploiting, in this way, the additional energy developed by the extension of the legs which pushes on fixed footrests. The rowers should try to maximize the force in the course direction. However, additional forces and moments in vertical direction and around the pitching axis (normal to the boat symmetry plane) are always introduced. A good rower should be able to minimize the parasitic motions that are induced by these additional forcing terms.

To fully appreciate the complexity of the system, one has to consider the interaction between the composite kinematics occurring on board with the hydrodynamics forces acting on the hull and the oar blades.

Even if the design of rowing hulls has a very long lasting tradition, manufacturers are still striving to improve their designs. As discussed in Chapter 3, mathematical models and advanced numerical methods can represent an effective tool that can be used to achieve performance improvements.

The earliest mathematical approach to rowing dates back to the twenties when the physical aspects of the problem were investigated [Ale25]. More recently, several models for the rowing boat dynamics, characterized by different levels of complexities. have been proposed In [Mil87], a simple model of the motion of a single-scull rowing hull is described. The motion of the rower within the boat is ignored and the hydrodynamic forces acting on the hull are derived from experimental data. The motion of the rowers was included in the model proposed in [BdM96], for an eight-oared shell, with the hydrodynamic forces still estimated on the basis of experimental measurements. In [SN93] and [Laz97], the wave component of the Resistance was estimated using Michell's thin ship theory [Mic98] and the viscous component using empirical correlations based on Reynolds similitude. To our best knowledge, the simulation of the free-surface viscous flow around a rowing boat using RANS-based methods has never been addressed in the literature.

Recently, a relevant research activity in this field has been carried out at the Modelling and Scientific Computing (MOX) laboratory of Politecnico di Milano (Italy) in collaboration with an important rowing shell manufacturer. In this context, a dynamic model which reconstructs the complex behaviour of a rowing boat in racing conditions has been developed [FM04]. Moreover, this model has been coupled and with different numerical models (viscous and inviscid) for the simulation of the flow around the hull [Con04, Pan04, DG04].

The full coupling between a RANS fluid solver and a dynamic model for the body motion has already been addressed in the literature (see, *e.g.*, [Azc01, Azc02a, ACS03]). These works mainly considered the prediction of the running attitude of a ship driven by a constant propulsion. The problem at hand involves a more complex dynamics due to the unsteady forces associated with the oar thrust and with the inertial effects given by the composite kinematic of the system. The solution of the complete fluid-structure interaction problem is beyond the objective of this work. Preliminary investigations on this subject have been presented in [Pan04].

In Olympic rowing, there exist 14 different boat classes. Eight of them participate to *sculling* races in which two oars are used, one in each hand. The other six classes participate to *sweep-oared* races in which the rower uses one oar with both hands. The sculling boat classes are the single, the double and the quadruple sculls with crews of one, two or four athletes respectively. The sweep row classes include the pair, the four, the lightweight four (for men only) and the eight with coxswain.

In this chapter, we present the results of numerical simulations that have been carried out to

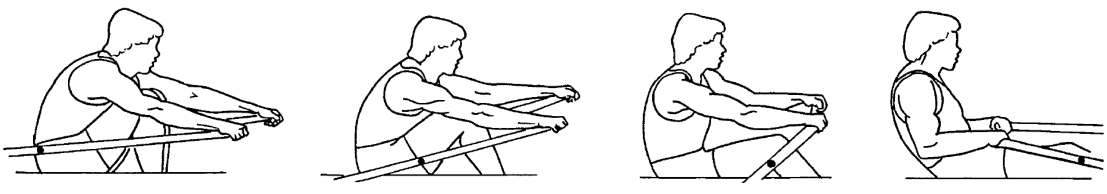


Figure 5.1: The four phases in a rowing cycle (from left to right): catch, drive, finish and recovery.

investigate the flow around a quadruple sculling boat. The numerical scheme described in Chapter 2 is used. Numerical results are compared with the available experimental data. We consider both steady and time-dependent boat attitudes and we analyse the role of the boat motion on drag estimation. Finally, we propose a possible design modification finalized to improve the performances of the boat and we present the some numerical simulations that have been carried out to assess its effectiveness.

5.2 Numerical simulations on steady configurations

In this section, we consider the free-surface flow around a rowing boat in steady motion. We first assess the accuracy of the numerical method described in Chapter 2 comparing the results of the simulation with available experimental measurements. Then, different hull designs are considered, and a preliminary estimation of the relative performances is given.

5.2.1 Comparison with experimental results

Experimental tests on a full scale rowing hull with different velocities and sink conditions have been carried out in the towing tank of the Istituto Nazionale per gli Studi ed Esperienze di Architettura Navale (INSEAN), in Roma, Italy. The towing tank measures $220\,\text{m} \times 9\,\text{m} \times 3.8\,\text{m}$ in longitudinal, transversal and vertical directions.

A first set of simulations has been performed with a computational domain which fits the geometry of the experimental towing tank. Due to the symmetry of the problem with respect to the mid plane of the boat, only one half of the geometry is considered. The computational domain is 70 m long, 4.5 m wide and 3.8 m deep. To simplify the grid generation, the upper side of the boat has been extended till the top side of the domain, as in the Wigley hull test case. A sketch of the computational domain and the reference system adopted are presented in Fig. 5.2

The boundary conditions used are analogous to those adopted in Section 4.2. However, on the lateral side of the domain a moving wall condition was imposed in order to better reproduce the presence of the wall in the experimental setup. The inflow velocity is 5.3 m/s, corresponding to $\text{Re} = 6 \times 10^7$ and $\text{Fr} = 0.48$, and is directed in in negative x-direction.

In this case, a block structured approach has been adopted for the grid generation. The computational domain is subdivided in sub-domains (called *blocks*) which are topologically equivalent to

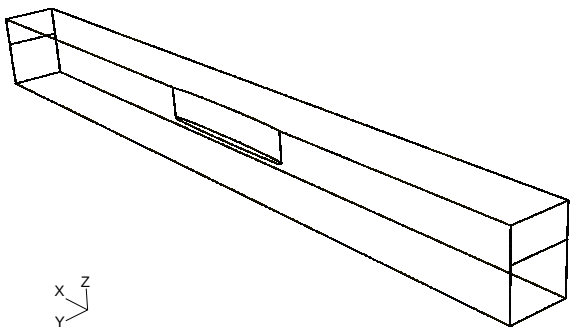


Figure 5.2: Computational domain and reference system for the rowing hull simulation.

	Total cell number	Nodes (x)	Nodes (y)	Nodes (z)
Grid 1	1254400	200	112	56
Grid 2	2450000	250	140	70
Grid 3	4750200	312	175	87

Table 5.1: Sizes of the three grids for the INSEAN test case.

cubes. In each block a structured grid is generated and each block grid matches at the interface the adjacent ones such that the grid is conformal.

Two regions of node clustering have been used in the computational grid to resolve the boundary layer around the boat and the water-air interface region. The grid spacing in the boundary layer depends on the turbulence model (realizable $k-\epsilon$) and on the wall-function used. In this respect, the same considerations made in Section 3.4 hold.

To assess the spatial convergence of the numerical scheme three grids with different refinements have been considered. The dimensions of the three grids are presented in Table 5.1 and a view of the middle size grid is displayed in Fig. 5.3. The simulations were run on 8 processor of the Linux cluster described in Section 2.4. The CPU times required to reach convergence varies between 6 hours for the small size case and 25 hours for the larger case.

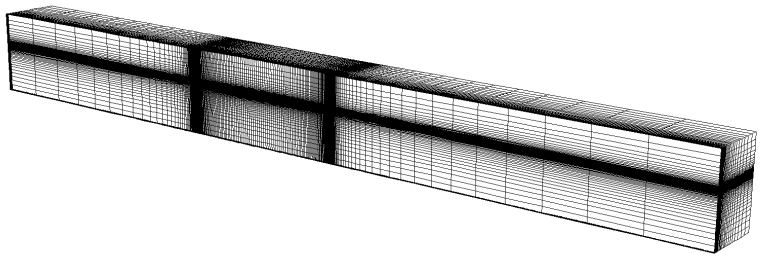


Figure 5.3: Perspective view of the grid for the INSEAN test.

A comparison of the drag on the different grids is reported in Table 5.2. These results show spatial convergence as the grid is refined and a global good agreement with the experimental measurements for the two finer grids. The grid convergence can also be appreciated observing the wave profiles on the hull, obtained with the three grids (see Fig. 5.4).

The wave patterns obtained on the three grids are displayed in Fig 5.5. We can note the perturbations on the wave field given by the presence of the lateral walls. In the simulation that will be presented in the following of this chapter, where the fitting of the towing tank geometry in no more required, a larger computational domain will be considered, in order to avoid this effect.

	Pres.	Visc.	Total	Error (%)
Experiments			267.0	
Grid 1	22.9	204.1	227.0	14.9
Grid 2	16.6	241.8	258.4	3.15
Grid 3	15.9	247.9	263.8	1.20

Table 5.2: Drag (N) comparison between experimental results and numerical computations over three different grids.

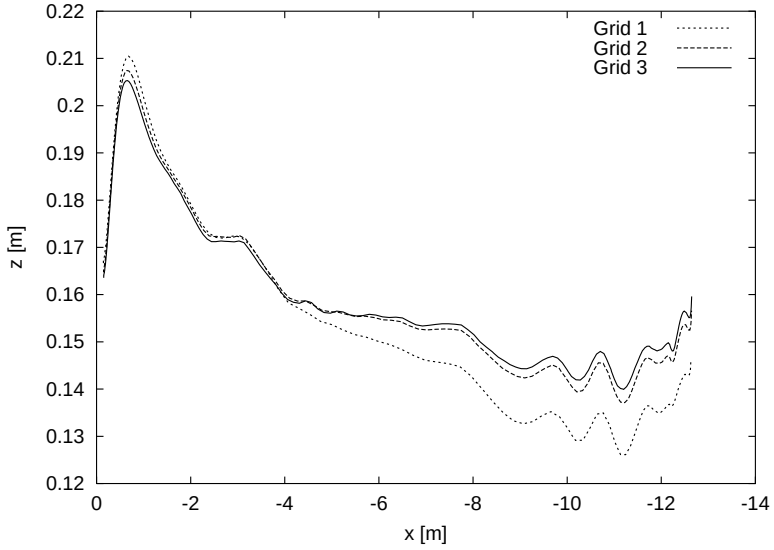


Figure 5.4: Comparison between the wave profiles on the hull obtained with the three grids.

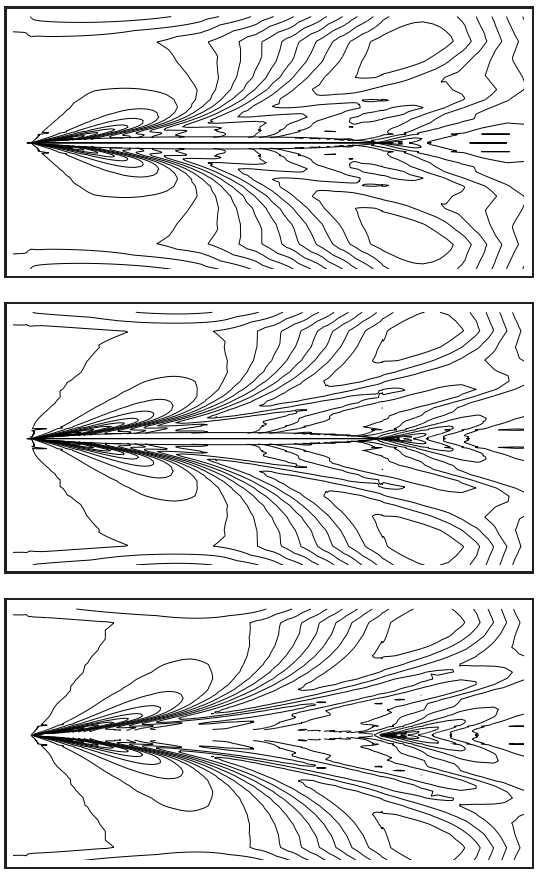


Figure 5.5: Comparison between the wave patterns obtained with the three grids: iso-line of free-surface elevation for (from top to right) Grid 1 (coarse), Grid 2 (medium), Grid 3 (fine)

5.2.2 Different hull designs

A second set of steady numerical simulations has been carried out to compare four different hull geometries. This investigation was finalized to supply a preliminary evaluation of the performances of different hull designs. A rowing boat typically undergoes an unsteady and complex motion during the race. (A characterization of the this motion and the numerical models that can be used to reproduce it will be described in the next section.) Therefore, if the objective of the study is the optimization of the boat performances in racing condition, the steady approximation can be considered too restrictive. However, preliminary indications on the behaviour of a given hull geometry can be extracted from steady simulation (as well as from "steady" towing tank experiments).

The free-surface flow around the four hulls have been simulated at three different pitching attitudes, namely the pitch angle β_P is set to $\beta_P = -1.5°, 0°, 1.5°$, respectively. For the reference system considered (see Fig. 5.2), a positive pitching attitude corresponds to have the bow lower than the stern. The inflow velocity is set to 5.3 m/s. To reduce the influence of the lateral sides, we have used a computational domain which is larger (15 m) than the one used in for the steady state simulations, keeping the same dimensions in x and z directions.

The four hulls have different volume distributions in longitudinal direction which result in different load distributions. In Fig. 5.6, the vertical component of the wall stress on the symmetry plane of the four hulls is displayed (for the case $\beta_P = 0°$).

We can note that Hull1 presents a stronger asymmetry, with respect to center of the boat, in the vertical (hydrostatic and hydrodynamic) load distributions. In particular, Hull1 is more loaded on the forward half that on the back one. We will see how this asymmetry will play a role in the analysis of its behaviour.

To measure the relative performances of the different hull designs, the simpler index that can be considered is the total drag acting on the hull. However, a measure of the pure drag does not give any indication about the tendency of the hull to emerge from the water when velocity increases, since the vertical force does not enter in the index.

A rowing boat should be able to reach a condition of planing on the water during the phase of maximal thrust. Indeed, the planing condition is typical of those high speed boat (usually referred to as *dynamically supported craft* having enough thrust to rise up on top of the water. It is well known that, in such condition, the upper velocity bound present in standard hulls when the Froude number increases is overcome (see *e.g* [ACS03]). Usually, the design of a rowing hull tries to maximize the planing effect. However, the possible modifications on the hull geometry are limited, on the one hand, due to the displacement constraints given by the presence of the rowers on board, on the other hand, because increasing the wetted surface increases the viscous component of the drag (which is the dominant one).

Based on this considerations, the *efficiency* of the hull, defined as the ratio between vertical force and drag, can be considered as a better indicator on the global performances. Drag and efficiency values for the four hulls at the three different pitching attitudes are reported in Figs 5.7. Indeed, it can be noted that an effective ranking between the four configurations can be obtained only when efficiency is considered. In particular, we note that the stronger asymmetry in the vertical load for Hull1 has a beneficial effect in terms of vertical lift and, therefore, in terms of efficiency.

5.3 A reduced model for rowing boat dynamics

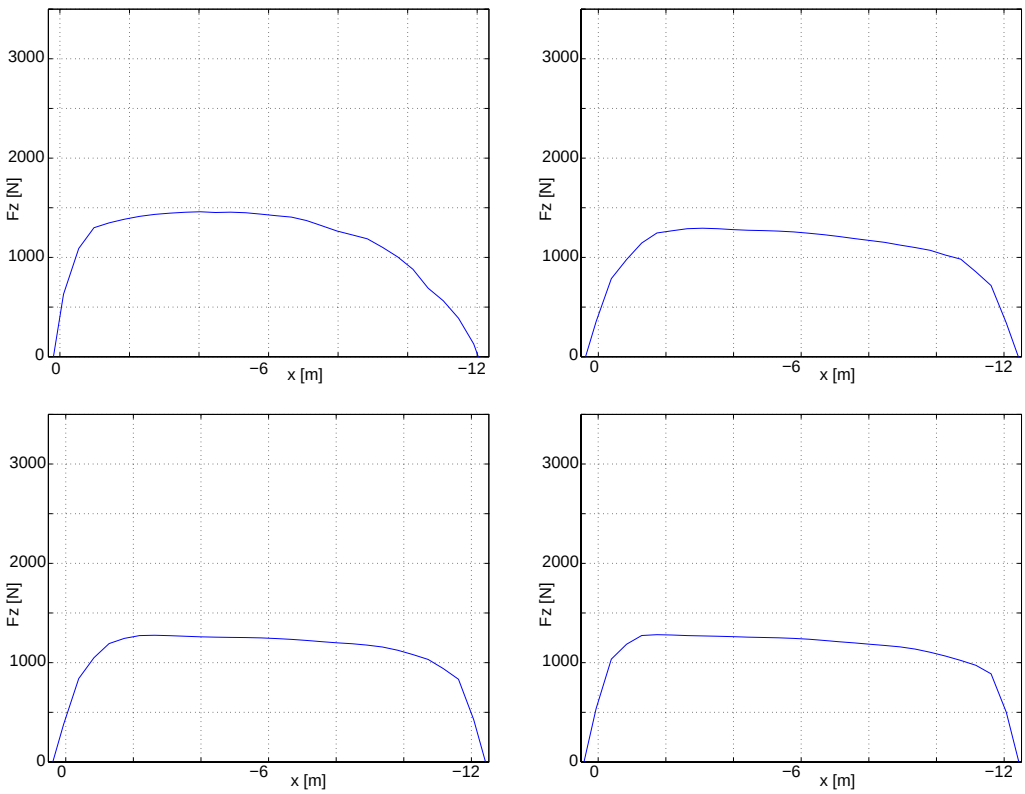
In this Section, we move toward a more complete investigation of the dynamic behaviour of the rowing boat due to the periodic movement of the rowers during the race.

The mathematical model describing the behaviour of a rowing boat in racing conditions is rather complex. Indeed, the different interactions between the various components (rowers, seats, oars, oarlocks, footrests and hull) should be taken into account in order to reproduce the complete dynamics of the system.

We consider the simplified dynamic model that have been developed at the MOX laboratory of Politecnico di Milano [FM04]. This model is based reconstructs the kinematic of the rowers/oars/hull system and has been used in the present study to simulate the unsteady behaviour of the boat with an imposed motion. A brief description of the simplified dynamic model is presented here, while for a complete description we refer to [DG04].

The complete dynamics of the rowing boat is simplified assuming that the hull is rigid and the motion is restricted to the symmetry (xz) plane of the boat. The forces acting on the hull can be treated as localized forcing terms. The position where each of them acts during the stroke has been determined based on the kinematic of the rowers/oars/seats system.

In the model, the intensity and direction of the forces exerted by seats, oarlocks and footrests on



vspace2mm

Figure 5.6: Distributions of the vertical component of the wall stress on the symmetry plane of the four designs: Hull1 (top-left), Hull2 (top-right), Hull3(bottom-left) and Hull4 (bottom-right).

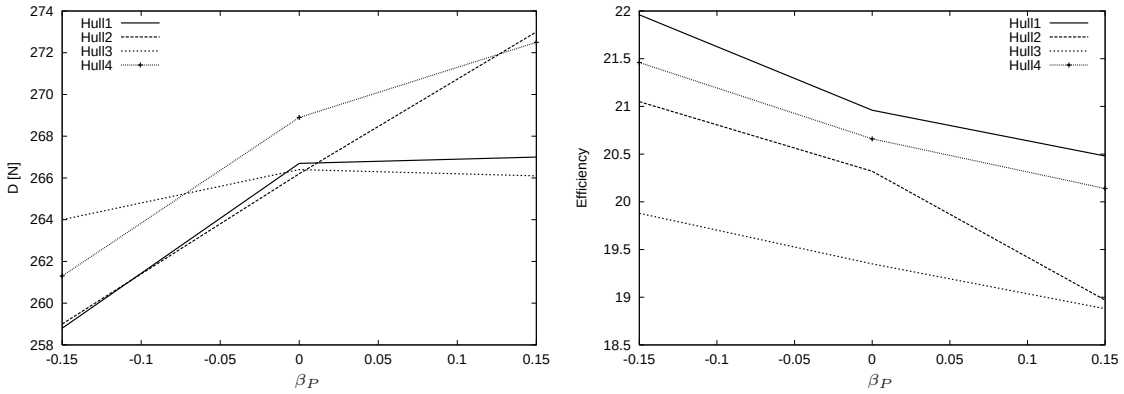


Figure 5.7: Drag (N) (left) and efficiency (right) for the four hull designs at three different pitching attitudes.

the hull have been estimated based on experimental measurements. The hydrodynamic forces are computed using a fast potential flow model. A scheme showing the different components of the system is presented in Fig. 5.8.

Once all the forcing terms, as well as the interactions between the different components, have been suitably modelled, a system of ordinary differential equations governing the dynamics of the hull can be obtained. This system reads

$$m_b \ddot{\boldsymbol{x}}_b = \boldsymbol{F}_b, \tag{5.1}$$

$$I_b \ddot{\theta}_b = M_b, \tag{5.2}$$

where m_b is the boat mass, $\ddot{\boldsymbol{x}}_b$ is the linear acceleration of the boat in the xz plane, $\boldsymbol{F}_b$ is the total force acting on the boat, $\ddot{\theta}_b$ is the pitching acceleration, I_b is the moment of inertia around the pitching axis and M_b is the total pitching moment acting on the boat. Equations (5.1) and (5.2) reflect the Newton's second law for forces and moments, respectively.

The contributions to the total force $\boldsymbol{F}_b$ and to the total moment M_b acting on the boat are given by:

$$\boldsymbol{F}_b = m_b \boldsymbol{g} + \boldsymbol{F}_H + \sum_i \boldsymbol{F}_i, \tag{5.3}$$

$$M_b = M_H + \sum_i (\boldsymbol{x}_i - \boldsymbol{x}_G) \times \boldsymbol{F}_i \tag{5.4}$$

where $\boldsymbol{F}_i$ are the forces applied by seats, oarlocks and footrests on the hull, $(\boldsymbol{x}_i - \boldsymbol{x}_G)$ are the moment arms of these forces, $\boldsymbol{x}_G$ is the position of the center of gravity, $\boldsymbol{F}_H$ and M_b are, respectively, the hydrodynamic force and pitching moment.

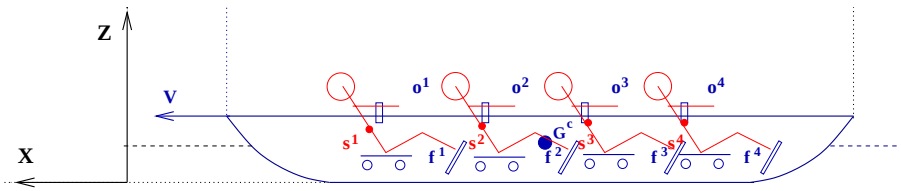


Figure 5.8: Schematic representation of the system showing the location of the forces acting on the hull ((s) seats, (o) oarlocks (f) footrests).

5.4 Numerical simulations with imposed pitching motion

The simplified dynamic system described in the previous section has been used to determine an approximated kinematic of the rowing boat in order to set up unsteady RANS computations with a prescribed boat motion. The objective of this investigation is to supply a first estimate of the drag component associated to the unsteady behaviour of the boat. Moreover, in the following, we will see how such approach can be used to evaluate the potential benefits of innovative design choices.

In the dynamic model, the time evolution of the force produced by each rower during a rowing cycle (see Fig. 5.9, left) is modelled as follows

$$F(t) = \begin{cases} A\,(\sin(\pi t/\tau_1))^2, & 0 \le t \le \tau_1, \\ 0, & \tau_1 \le t \le \tau_2 \end{cases}$$

where $\tau_1 = 0.71$ s is the duration of the stroke phase, $\tau_2 = 0.80$ s is the duration of the recovery phase and the peak of the stroke force is $A = 1200$ N. The total period of a rowing cycle $T = \tau_1 + \tau_2 = 1.51$ s corresponds to a typical race frequency of 39 strokes per minute.

The time evolution of the pitching angle, obtained from the solution of system (5.2), is presented in Fig. 5.9 (right). This result has been used to set up a time dependent simulation of the flow around a rowing boat which rigidly rotates around its pitching axis. The motions along x and y directions are neglected.

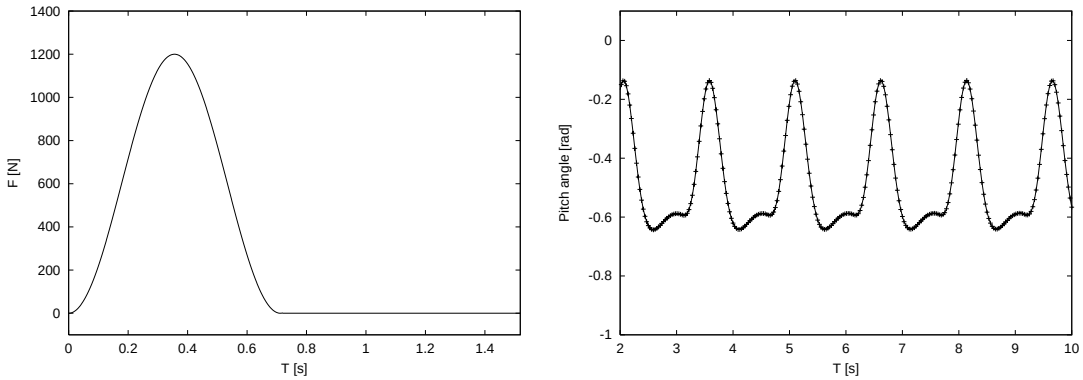


Figure 5.9: Time evolution of the force produced by each rower (left) and correspondent pitching angle time evolution.

5.4.1 Moving grid approach

To take into account the pitching motion of the boat in the solution of the RANS equations different strategies are possible. We adopted an approach based on a *sliding grid* technique. The computational grid is decomposed in two or more subdomains which rigidly move one with respect to the other, along the common interface. Two examples of rotational and translational sliding grid setups are displayed in Fig. 5.10.

To take into account the grid motion, the conservation laws introduced in Chapter 2 for momentum, volume fraction and turbulence quantities have to be suitably modified. In particular, the Navier–

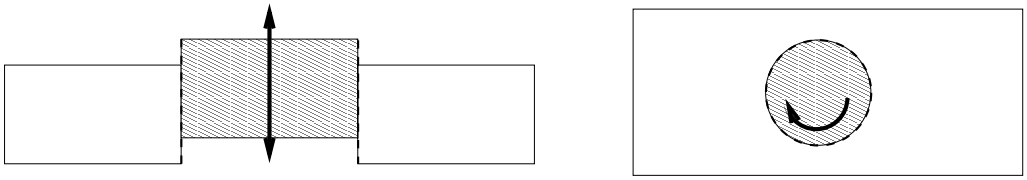


Figure 5.10: Translational (left) and rotational (right) sliding grid examples. The shaded regions are the moving portions of the domain and red lines identify the interface zones.

Stokes system (2.1)-(2.2) written in conservation form, becomes

$$\int_c \partial_t(\rho \boldsymbol{u}) dV + \oint_{\partial c} \rho\,(\boldsymbol{u}-\boldsymbol{w}) \otimes \boldsymbol{u}\cdot\boldsymbol{n}\, d\Gamma - \oint_{\partial c} \boldsymbol{T}(\boldsymbol{u},p)\cdot\boldsymbol{n}\, d\Gamma = \int_c \rho \boldsymbol{g} dV, \tag{5.5}$$

$$\oint_{\partial c} \rho\, \boldsymbol{u}\cdot\boldsymbol{n}\, d\Gamma = 0, \tag{5.6}$$

where $\boldsymbol{w}$ denotes the velocity field which define the grid motion. An analogous modification in the convective term of the volume fraction and turbulence equations has to be included [Flu03].

As the grid moves, the mesh conformity at the interface between the sub-domains is no more guaranteed. As a consequence of the grid motion, a suitable strategy, which takes into account the actual relative position of the grid elements, has to be adopted to compute the fluxes through the element boundaries.

In the example shown in Fig. 5.11, we consider the boundaries of two sub-domains composed of faces A-B and B-C, and faces D-E and E-F, respectively. The intersection of these zones defines a sequence of boundary sections which are denoted as *a-d*, *d-b*, *b-e*, *e-c and c-f*. To compute the flux across the interface into cell 4, for example, the fluxes computed on sections *d-b* and *b-e* are used, bringing information into cell 4 from cells 1 and 3, respectively. The portions of the sub-domain boundaries where the two sub-domains do not overlap (faces *a-d* and *c-f*, in the example) need a special treatment. In our case, inflow and outflow boundary conditions have been imposed according to the local flow direction.

In our case, the computational domain around the rowing boat has been partitioned in three sub-domains (see Fig. 5.12). The central one contains the boat that will rotate around the pitching axis passing through the center of gravity of the boat according to the pitching angle time evolution that has been determined. A suitable block structured mesh for this domain has been created. In Fig. 5.13 the blocks that define the grid are displayed. The grid has 250 nodes in x direction, 170 nodes in y direction and 70 nodes in z direction.

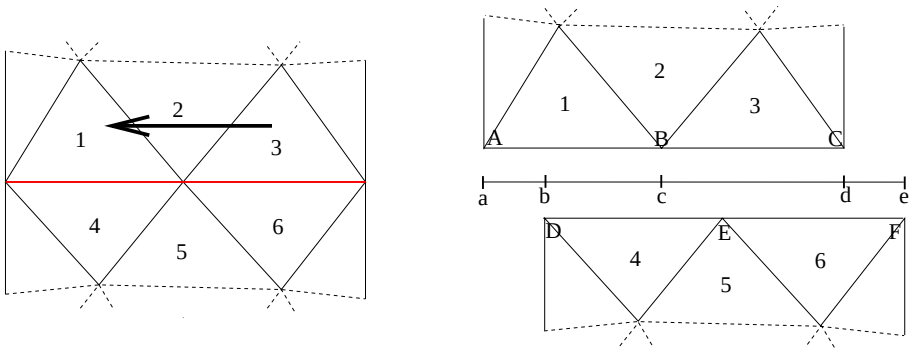


Figure 5.11: Flux treatment of the interface regions in moving grid problems.

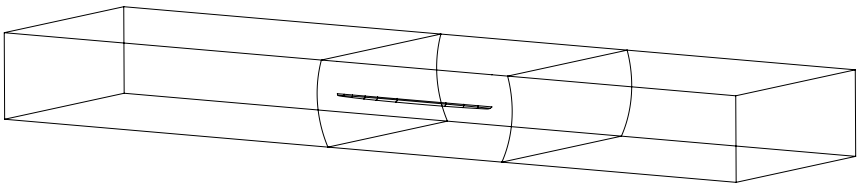


Figure 5.12: Computational domain for the moving mesh simulation of the rowing boat.

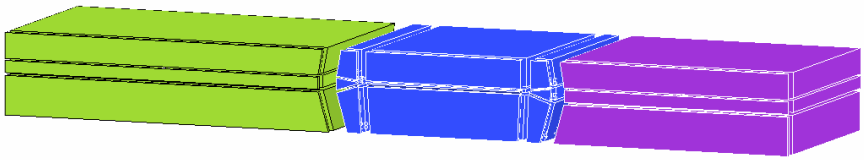


Figure 5.13: Mesh blocking the moving mesh simulation of the rowing boat.

5.4.2 Numerical results

To estimate the effect of the boat dynamics on its performances two simulations, one steady and one with the imposed pitching motion, have been performed. The time dependent simulation has been initialized with the steady state solution and it has been run, using a time step $\Delta t = 0.01$ s, till the time evolution of the forces was stabilized. We used a time step $\Delta t = 0.01$ s. The periodical time evolution of the drag the hull obtained in the time dependent simulation is presented in Fig. 5.14.

In Table 5.3, we present a comparison between the drag acting on the hull in the steady configuration and the average over the period of the drag computed in the unsteady case. It can be noted that the pitching motion (which is only one of the movements that a rowing boat undergoes in its real dynamics) increases the total drag of about 10%. In particular, , as it could be expected, it contributes mostly on the pressure component of the drag. Indeed, the pressure drag is more than doubled with respect to the steady state simulation. The evolution of the free-surface around the bow of the hull at different time instants covering one time period is presented in Fig. 5.15.

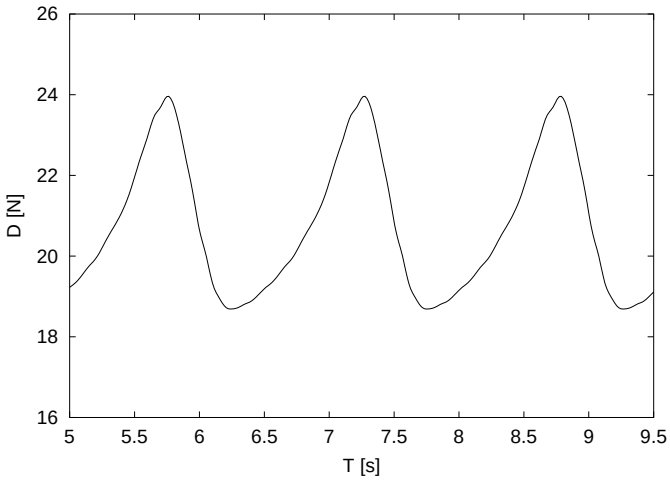


Figure 5.14: Time evolution of the drag on the hull for the imposed pitching motion case.

	Pres.	Visc.	Total
Steady state	15.6	241.3	256.9
Imposed pitching motion	37.7	248.2	285.9

Table 5.3: Drag [N] components for the steady state case and he imposed pitching motion case.

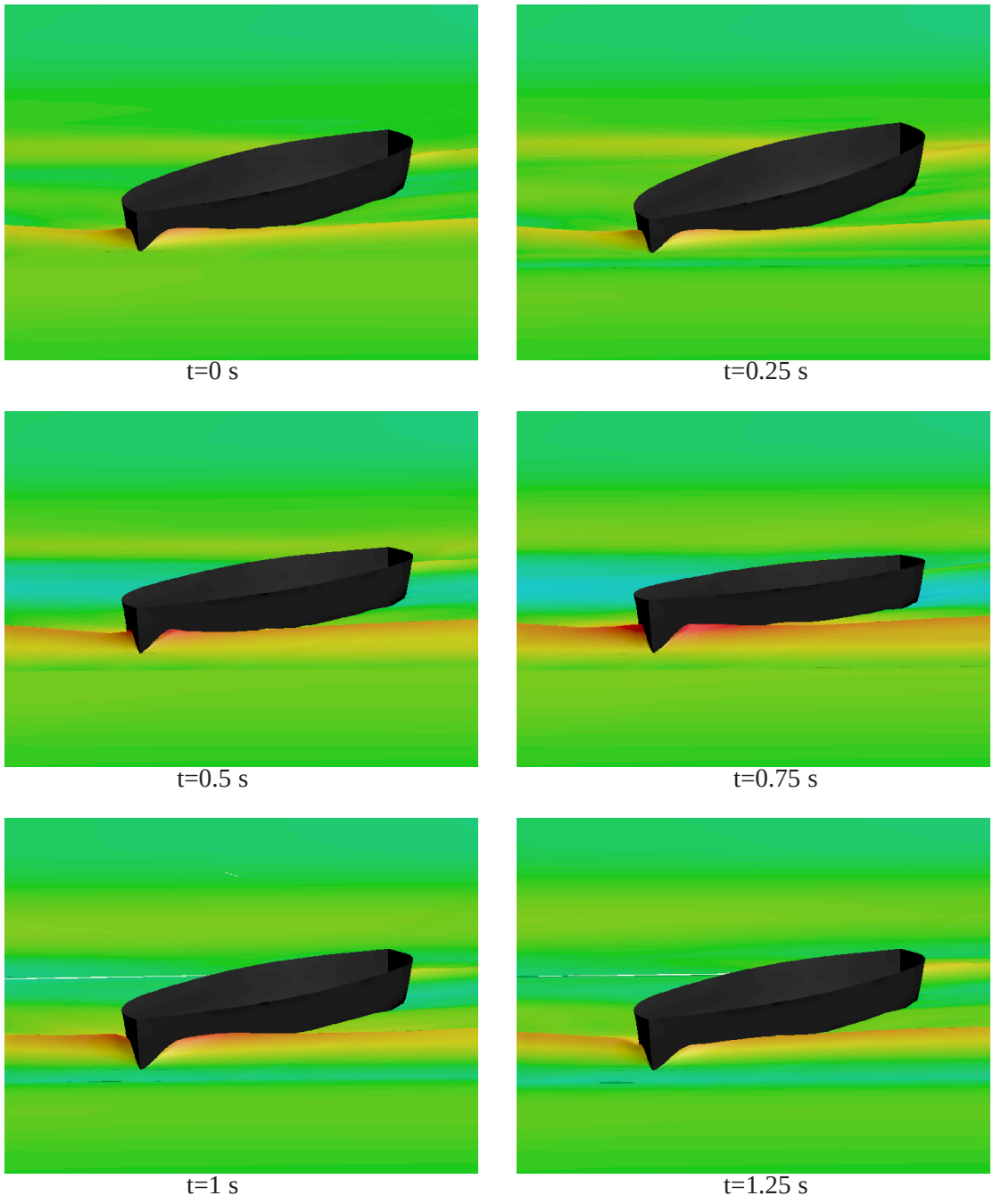


Figure 5.15: The evolution of the free-surface around the bow of the hull at different time instants.

5.5 Introduction of pitch stabilizers

The pithing dynamics has been found to have a relevant effect on the total drag acting on the rowing hull. Although the pitching motion is only one of the components that characterize the full dynamics of the boat, a better understanding of its effect and the introduction of design concepts to reduce its negative impact could contribute to an enhancement of the global boat performances.

One possible approach to this problem is the adoption (allowed by the Olympic rules) of stabilizing wings on the sides of the hull. These appendices could reduce the pitching motion and, therefore, the associated drag component.

In this Section, we will describe how suitable appendages can be designed and we will present numerical results which assess the potential improvement deriving from their use.

5.5.1 Design of the pitch stabilizer

The first step to be addressed is the hydrodynamic design of the appendices, that is the definition of their geometrical characteristics and their positioning on the hull. A preliminary consideration can be done on their longitudinal positioning. First of all, they should be placed as far as possible from the center of rotation in order to maximize their moment arm. The two possible choices are positioning them either in the bow region or in the stern region.

If we consider a stabilizer with a symmetric profile and with zero angle of attack (in the average pitching attitude of the boat), the bow and stern positions have opposite behaviours in term of stability. Indeed, as sketched in Fig. 5.16, the stabilizers in stern position are stable, since a perturbation on the pitching angle induces a force on the stabilizer which acts to contrast the perturbation itself. On the other hand, the bow position is unstable.

Although the stern position could appear the natural choice, other factors should be taken into account. Any force acting in negative vertical direction should be avoided for reasons related to planning effects. It should be noted that, in the case at hand, an unstable behaviour is not completely negative. In fact, if on the one hand, we want to reduce the pitching motion, on the other hand, we want to encourage a negative pitch attitude (with the bow higher than the stern). By choosing adequately the profile, as well as the geometrical angle of attack, it is possible to

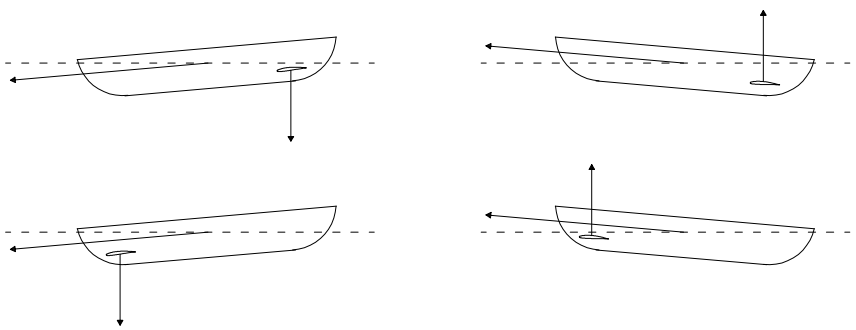


Figure 5.16: Stability of the pitch stabilizer: stable stern position (top) unstable bow position (bottom).

obtain a *semi*-stable configuration with the stabilizers in bow position. With *semi*-stable, we mean that the stabilizer will be stable for positive pitching velocity (if the bow tends to going down the stabilizer acts in the opposite direction) and unstable for negative pitching velocity. In this way, we can guarantee that the vertical component on the stabilizer in always in positive direction.

In particular, we have used a H105 hydrofoil [Spe01] with a geometrical angle of attack of 2 degrees which guarantees a positive lift over the range of incidences that the stabilizer will undergo. The H105 hydrofoil has been preferred to other standard hydrofoils for both hydrodynamic (C_D/C_L polar, cavitation and laminarity) and structural considerations (thickness).

The dimensions of the stabilizer have been determined taking into account the construction constraints and estimating the pitching moment that they should be able to produce during the stroke cycle. The stabilizer has a chord of 6 cm and a span of 36 cm, corresponding to an aspect ratio $\Lambda = 6$. Ideally, the design should also prescribe an optimal sweep and dihedral angle, as well as a twist angle distribution. The analysis that is presented here is restricted to a stabilizer with dihedral angles $\alpha = 15°$, no sweeping and no twisting. The stabilizer is placed at 1.2 m from the bow and at 20 cm under the hydrostatic water level. The configuration is displayed in Fig. 5.17

To assess the possible benefits deriving from the introduction of the pitch stabilizers, we have proceeded as follows:

1. the forces acting on the stabilizer during one pitching period have been estimated simulating the flow around the isolated stabilizer;

2. then the contribution of these forces on the dynamics of the complete boat is estimated taking them into account in the dynamical system;

3. finally, the new pitching kinematic obtained from the dynamical system is used to simulate the flow around the whole boat with the stabilizer attached.

This approach consists in just one feedback loop which represents a coarse approximation of the real effect of the stabilizer on the complete dynamics. However, it allows us to give a first estimate of the behaviour of the system associated to the proposed design modification.

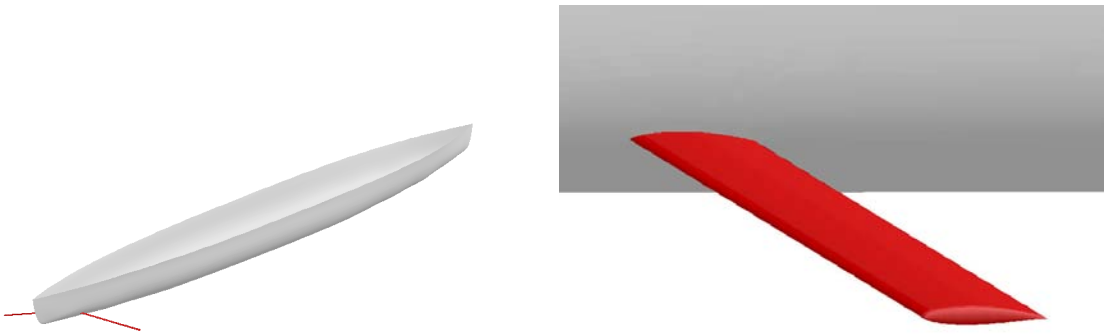


Figure 5.17: Positioning of the pitch stabilizers on the hull (left) and a detailed view of its geometry (right).

5.5.2 Forces on the isolated stabilizer

To estimate the forces due to the stabilizer, we have simulated the flow around the isolated stabilizer subjected to an imposed motion.

Given the rigid pitching rotation $\beta_P(t)$ of the hull around its center of gravity, the motion of the isolated stabilizer can be obtained as the superposition of a local vertical translation v_z^T and a local rotation $\boldsymbol{w}^R$ around the center of the stabilizer. Considering that the oscillations of the pitching angle are small ($< 1°$), the velocity of each point $\boldsymbol{x}$ on the surface of the stabilizer can be approximated as follows:

$$\boldsymbol{u}_s(\boldsymbol{x},t) \approx v_z^T(t)\boldsymbol{k} + \boldsymbol{w}^R(\boldsymbol{x},t), \tag{5.7}$$

where

$$\begin{aligned} v_z(t) &= (\boldsymbol{x}_{G,s} - \boldsymbol{x}_G) \times \dot{\beta}_P \boldsymbol{j}, \\ \boldsymbol{w}^R(\boldsymbol{x},t) &= (\boldsymbol{x} - \boldsymbol{x}_{G,s}) \times \dot{\beta}_P \boldsymbol{j}, \end{aligned}$$

$\boldsymbol{j}$ and $\boldsymbol{k}$ being, respectively, the unit vectors in y and z directions. The points $\boldsymbol{x}_{G,s}$ and $\boldsymbol{x}_G$ are the centers of gravity of stabilizer and hull, respectively.

Using this decomposition, we have set up a simulation of the water flow around the isolated stabilizer, resorting to the same sliding grid approach described above. In particular, in this case, the computational domain has been split into three sub-domains: a cylindrical region containing the stabilizer which rotates with angular velocity $\boldsymbol{w}^R$, a parallelepipedal region containing the cylinder which translates vertically with velocity v_z^T and a fixed external sub-domain (see the two-dimensional sketch in Fig. 5.18).

The computational domain measures 2 m in x direction, 1 m in y direction and 1 m in z direction. A block structured grid has been used. In Fig. 5.19, the three dimensional domain and the grid blocks are presented. A particular of the grid around the stabilizer is shown in Fig. 5.20.

First, a steady solution on the fixed geometry has been computed. Then the translational and rotational grid movements defined by equation (5.7) were activated and the time dependent simulation was run until the forces assumed a periodic behaviour. The time evolutions of drag and lift components on the stabilizer are presented in Fig. 5.21.

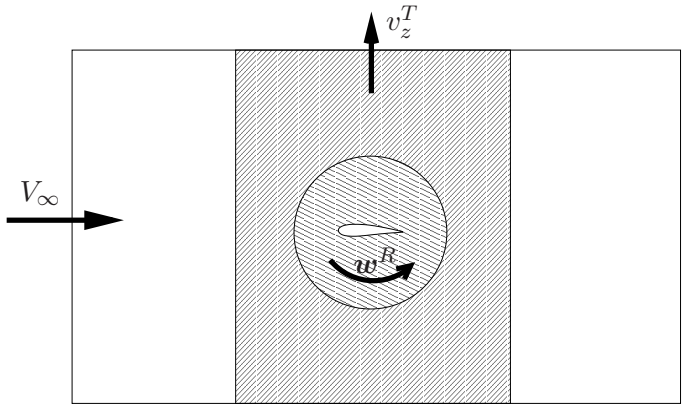


Figure 5.18: Moving domains for the isolated stabilizer.

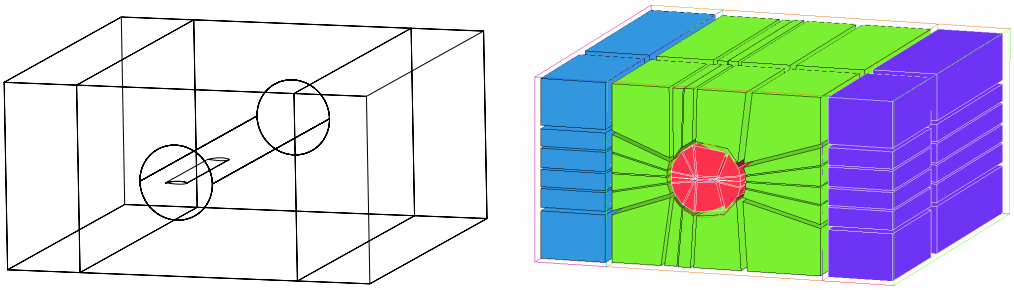


Figure 5.19: Computational domain (left) and grid blocks (right) for the isolated stabilizer.

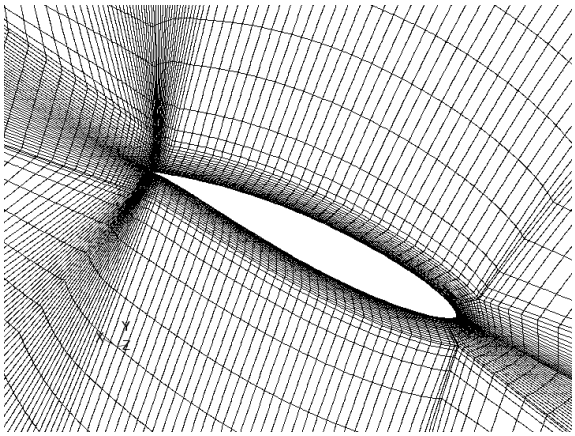


Figure 5.20: Detail of the computational grid around the stabilizer.

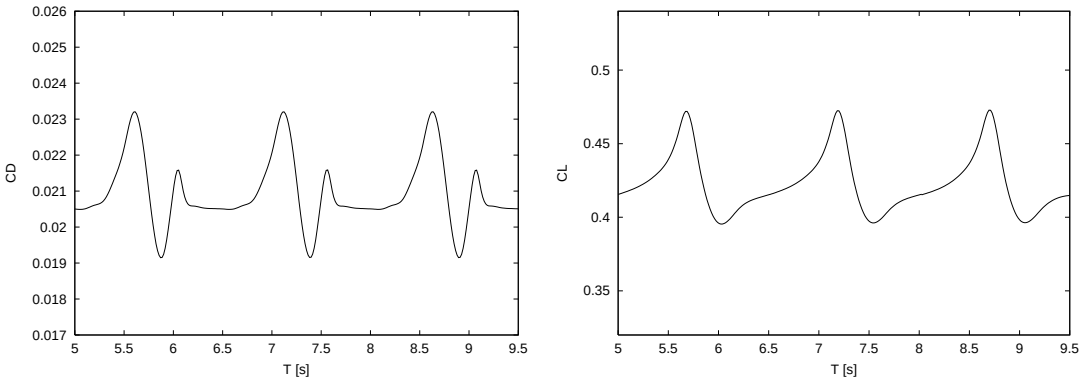


Figure 5.21: Time evolution of the drag (left) and lift (right) coefficients on the moving stabilizer.

5.5.3 Effect of the pitch stabilizer on the global performances

As discussed above, an appropriate design of the pitch stabilizer should be able to improve the global performance of the rowing boat. In particular, we expect that the stabilization effect on the pitching motion could reduce the component of the drag associated to this movement. This component has been estimated in Sec. 5.4 to be around 10% of the total resistance.

We have first performed a steady state simulation of the flow around the boat with the stabilizer in the bow region, in order to to analyse the impact on the "steady" performances induced by the introduction of this element. A fixed pitch angle $\beta_P = 0$ has been considered. In Fig. 5.22, the surface grid on the hull with the stabilizer is shown.

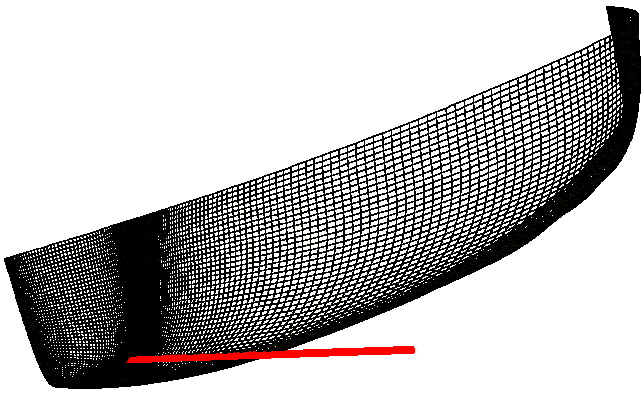


Figure 5.22: Surface grids on the hull with the stabilizer.

In Table 5.4, we present the comparison between the drags computed on the bare hull configuration and the one with stabilizer. Quite surprisingly, the configuration with the stabilizer gives a lower total drag. Even more surprisingly, the reduction is mainly due to the viscous component. This was not expected since the viscous drag, which is mainly related to the wetted surface, should always be penalized by introducing an additional component underwater.

Configuration	Pres.	Visc.	Total
Bare hull	15.6	241.3	256.9
Hull + Stabilizer	24.7	223.8	248.5

Table 5.4: Drag (N) components for the steady state cases with and without stabilizers.

However, observing Fig. 5.23, we can note that the wave profile on the hull is highly influenced by the presence of the stabilizer. In particular, we can note that the suction effect given by the low pressure on the top side of the stabilizer, induce a faster lowering of the bow wave profile. This effect results in a global reduction of the wetted surface. Moreover, we can observe that, in the

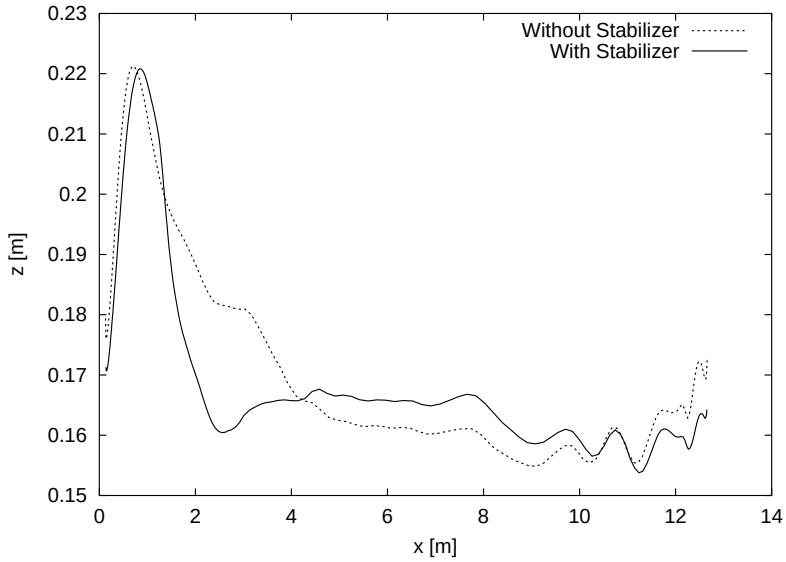


Figure 5.23: Wave profiles on the hull for the configurations with and without stabilizer.

presence of stabilizer, the peak of the wave bow occurs around 20 cm downstream with respect to the bare hull case.

In Fig. 5.24, we present the density distributions on the boat for the two configuration. We note that, the interface diffusion, already observed in the free-surface test cases discussed in Chapter 4, occurs in this case, as well.

We consider next the simulation of the flow around the same two configurations subjected to an imposed pitching moment. As discussed above, a new time evolution of the pitching angle has been computed by the dynamical system adding the contribution of the forces on the stabilizer, as they were evaluated in the previous section.

If Fig. 5.25, we report the time evolutions of the pitching angle obtained by the dynamical system with and without the force contribution given by the stabilizer. We can note the strong influence that the stabilizer has on the pitching dynamics. The amplitude of the pitching angle oscillations is reduced by almost 50%. This reduction by itself is already a good indication of the benefit that the stabilizer can give. It should be noted, on the other hand, that a collateral effect is the reduction (in modulus) of the mean pitching angle. This could represent a drawback since it could

Figure 5.24: Density distribution on the bare hull (left) and on the hull with stabilizer (right).

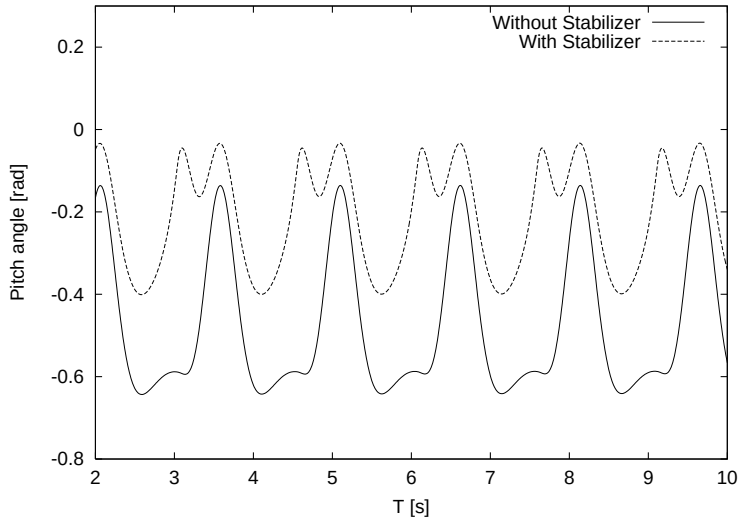


Figure 5.25: Comparison between the time evolution of the pitching angle for the cases with and without stabilizers.

affect the ability of the hull to reach a planing condition. To further investigate this aspect, a complete fluid-structure coupling, which accounts also for the motion in x and z directions, would be required.

Finally, the new time evolution of the pitching moment (Fig. 5.25) has been used to simulate the flow around the hull with the stabilizer attached given an imposed pitching motion. Again, a steady state solution is first obtained around a fixed position and then the pitching motion is activated.

The periodic time evolution of the drag for the two configurations is presented in Fig. 5.26. As expected from the analysis of the pitching angle time evolution, the mean value of the drag for the configuration with stabilizer is lower, as well as the amplitude of the drag oscillations.

In Table 5.5, we report a comparison between the drag on the two configurations for the steady state simulation and the mean value of the drag for the imposed motion cases. The configuration with stabilizer perform better in both steady and unsteady case. Moreover, we observe that the increase in drag associated to the pitching motion is consistently lower for the configuration with stabilizer.

From the results presented here, we can conclude that the design modification proposed has a beneficial effect on the overall performances of the boat. Clearly, to fully understand the potential improvement that could be achieved by the use of stabilizers in Olympic rowing boats further investigations are necessary.

In particular, the different design parameters defining the stabilizer (profile, angles of attack, sweep and twist, planform shape) should be investigated and, possibly, optimized. Moreover, the approach presented here does not account for the real fluid-structure interaction. A complete fluid-structure numerical simulation would be required to fully assess the impact that the stabilizer has on the boat performances.

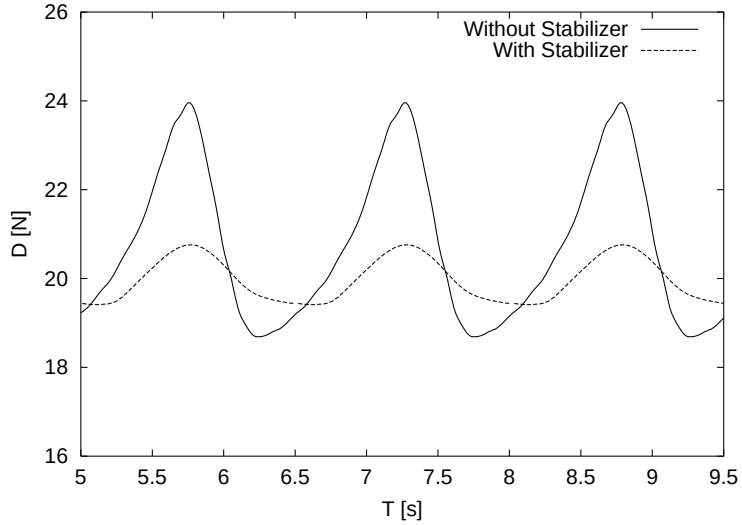


Figure 5.26: Comparison between the time evolution of drag for the cases with and without stabilizers.

	Steady	Time Dependent (average)	Difference (%)
Bare hull	256.9	286.6	11.1
Hull + Stabilizer	248.5	257.7	3.6

Table 5.5: Comparison between the drag (N) for the steady cases and the averaged value for the time-dependent cases with and without stabilizer.

Chapter 6

A Level Set Finite Element Method for the Solution of Free-Surface Flows

In the previous chapters, we have introduced a numerical method for the solution of viscous free-surface flows which has been applied to the simulation of free-surface problems in naval engineering applications. This method relies on a Volume of Fluid (VOF) technique (see Section 1.2.2) in combination with a finite volume spatial discretization. This approach is nowadays the standard in most of the RANS commercial solver with free-surface capabilities available on the market. The numerical results have been discussed in Chapter 5.

We have seen that the Volume of Fluid method can be used for the numerical simulation of free-surface problems with complex interface dynamics. However, as pointed out in Chapter 4 and 5 for different test cases, one limitation of this approach is related to the intrinsic numerical difficulties associated with the advection of a discontinuous characteristic function. In particular, unless suitable (but often too expensive in three dimensional problems) interface reconstruction techniques are adopted, severe numerical diffusion is introduced, leading to the smearing of the interface region over several grid element and consequently affecting the accuracy of the force estimations.

In this second part of the thesis, we will introduce and analyse a numerical method for the solution of viscous two-fluid flows based on the level set method, coupled with the solution of the incompressible Navier–Stokes equations, in a finite element framework. A short introduction of the level set method has been presented in Section 1.2.2, where the advantages and disadvantages of this approach when compared to the Volume of Fluid method have also been discussed.

The objective is to develop a general strategy for the solution of free-surface flow problems. The choice of finite elements for the spatial approximation of the problem has been motivated by several aspects: *(i)* their intrinsic capability to deal with complex geometries, *(ii)* the well established theoretical bases which allows rigorous analysis of the numerical schemes, *(iii)* the natural treatment of discontinuous coefficients in the variational formulation of the problem. As pointed out in Section 1.2.2, the use of level set methods in combination with finite element discretization has not yet been intensively explored and just few works have been proposed in the literature. We believe that this approach deserve more investigation and that the positive features of level set method and finite elements can be successfully combined to achieve accurate and effective numerical solutions of complex free-surface flows.

The numerical method and the simulations presented in the following chapters are restricted to

two-dimensional laminar flows. However, the different components defining the method (temporal and spatial discretizations, stabilization technique, reinitialization procedure) have been developed such that they easily can be extended to three-dimensional problems.

The techniques introduced in this thesis are currently being extended to three-dimensional problems in the framework of the library *LifeV*, a three dimensional finite element code developed in a joint collaboration between École Polytechnique Fédérale de Lausanne (CMCS), Politecnico di Milano (MOX) and INRIA (BANG).

6.1 Problem setting

We have seen in Chapter 1 that the density-dependent (inhomogeneous) Navier–Stokes equations (1.5)-(1.7) provide a suitable mathematical model to describe viscous two-fluid flows. Besides, in Section 1.2, we have discussed some possible numerical approaches for the approximation of this problem.

In the *Level Set method*, the interface is defined as the zero level set of the signed distance function from the interface Γ, designed to be positive in fluid 1 and negative in fluid 2. Given a known velocity field $\boldsymbol{u}$, the evolution of the interface is determined by solving the following advection equation for ϕ:

$$\phi_t + \boldsymbol{u} \cdot \boldsymbol{\nabla}\phi = 0. \tag{6.1}$$

The coefficients in the inhomogeneous Navier–Stokes equations (1.5)-(1.7) are functions of ϕ and can be defined as follows:

$$\begin{aligned} \rho(\phi) &:= \rho_1 + (\rho_2 - \rho_1)\, H(\phi), \\ \mu(\phi) &:= \mu_1 + (\mu_2 - \mu_1)\, H(\phi), \end{aligned} \tag{6.2}$$

where H is the Heaviside function.

The two-fluid flow problem in $\Omega \times [0, T]$ can therefore be described, combining system (1.5)-(1.7) with equations (6.1) and (6.2), by the following continuous model:

$$\rho(\phi)\,(\boldsymbol{u}_t + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}) - \boldsymbol{\nabla}\cdot\,(\mu(\phi)(\boldsymbol{\nabla}\boldsymbol{u} + \boldsymbol{\nabla}\boldsymbol{u}^T)) + \boldsymbol{\nabla}p \;=\; \rho(\phi)\boldsymbol{g} + \kappa\sigma\delta(\phi)\boldsymbol{n}, \tag{6.3}$$

$$\boldsymbol{\nabla}\cdot\,\boldsymbol{u} \;=\; 0, \tag{6.4}$$

$$\phi_t + \boldsymbol{u}\cdot\boldsymbol{\nabla}\phi \;=\; 0, \tag{6.5}$$

where δ is the Dirac function. Suitable boundary and initial conditions for $\boldsymbol{u}$ and ϕ must be imposed to close the problem. In the following, we consider homogeneous Dirichlet boundary condition on the velocity.

To rewrite problem (6.3) in its weak form, we first introduce the following functional spaces

$$\boldsymbol{V} \;:=\; (H_0^1(\Omega))^d = \{\boldsymbol{v} \in (H^1(\Omega))^d \mid \boldsymbol{v}|_{\partial\Omega} = 0\}, \tag{6.6}$$

$$Q \;:=\; L_0^2(\Omega) = \left\{ q \in L^2(\Omega) \mid \int_\Omega q = 0 \right\}, \tag{6.7}$$

$$W_{\boldsymbol{\beta}}(\Omega) \;:=\; \{\psi \in L^2(\Omega) \mid (\boldsymbol{\beta}\cdot\boldsymbol{\nabla}\psi) \in L^2(\Omega)\}, \tag{6.8}$$

$$W_{\boldsymbol{\beta}}^0(\Omega) \;:=\; \{\psi \in V_{\boldsymbol{\beta}}(\Omega) \mid \psi|_{\partial\Omega^-} = 0\}, \tag{6.9}$$

where $\boldsymbol{\beta}$ is a vector field such that $\boldsymbol{\nabla\cdot\beta} = 0$ and $\partial\Omega^- = \{\boldsymbol{x} \in \partial\Omega \mid \boldsymbol{\beta}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) < 0\}$. L^2 is the space of square integrable functions and $H^1 = \{v \in L^2(\Omega) \mid \boldsymbol{\nabla} v \in L^2(\Omega)\}$.

We denote by $(\cdot,\cdot)$ the L^2-scalar product on Ω

$$(f,g) := \int_\Omega f\, g\, dx, \qquad f,g \in L^2(\Omega)$$

and with $\|\cdot\|_0$ the associated norm

$$\|f\|_0 := (f,f)^{1/2}, \qquad \forall f \in L^2(\Omega).$$

$W_{\boldsymbol{\beta}}(\Omega)$ is a Hilbert space equipped with the norm $\|\psi\|_{1,\boldsymbol{\beta}} = (\|\psi\|_0^2 + \|\boldsymbol{\beta}\cdot\boldsymbol{\nabla}\psi\|_0^2)^{1/2}$.

We introduce the bilinear forms:

$$\begin{aligned} m_\phi(\boldsymbol{u},\boldsymbol{v}) &:= (\rho(\phi)\boldsymbol{u},\boldsymbol{v}) \\ a_\phi(\boldsymbol{u},\boldsymbol{v}) &:= (\mu(\phi)(\boldsymbol{\nabla u} + \boldsymbol{\nabla u}^T), \boldsymbol{\nabla v}), \\ b(\boldsymbol{u},q) &:= (\boldsymbol{\nabla\cdot u}, q), \\ l_{\boldsymbol{u}}(\phi,\psi) &:= (\boldsymbol{u}\cdot\boldsymbol{\nabla}\phi, w) \end{aligned}$$

and the trilinear form:

$$c_\phi(\boldsymbol{u};\boldsymbol{v},\boldsymbol{w}) := (\rho(\phi)(\boldsymbol{u}\cdot\boldsymbol{\nabla v}), \boldsymbol{w})$$

The weak formulation of problem (6.3) reads:

P6.1 *Find $\boldsymbol{u}(\boldsymbol{x},t) \in \boldsymbol{V}$, $p(\boldsymbol{x},t) \in Q$ and $\phi(\boldsymbol{x},t) \in W^0_\beta$ such that:*

$$\begin{aligned} m_\phi(\boldsymbol{u}_t,\boldsymbol{v}) + a_\phi(\boldsymbol{u},\boldsymbol{v}) + c_\phi(\boldsymbol{u};\boldsymbol{u},\boldsymbol{v}) - b(\boldsymbol{v},p) &= m_\phi(\boldsymbol{g},\boldsymbol{v}) + (\boldsymbol{f}_\Gamma,\boldsymbol{v}), && \forall \boldsymbol{v} \in \boldsymbol{V}, \\ b(\boldsymbol{u},q) &= 0, && \forall q \in Q, \\ (\phi_t, w) + l_{\boldsymbol{u}}(\phi,w) &= 0, && \forall w \in L^2(\Omega), \end{aligned}$$

for each $t \in]0,T]$.

6.2 Finite Element discretization

In this section, we introduce a spatial discretization of problem P6.1 based on a finite element approach. A theoretical analysis concerning the finite element approximation of Navier–Stokes equations with free-surface can be found in [TT01]. For a general review on numerical methods for the solution of the incompressible Navier–Stokes equations, we refer to [QV94, HR88].

We restrict our attention to the 2-dimensional case. Let us consider a triangulation $\mathcal{T}_H$ of the domain Ω. For the finite element approximation the Navier–Stokes system in problem P6.1, we consider the so-called ($\mathbb{P}_1$-iso$\mathbb{P}_2$)-$\mathbb{P}_1$ elements introduced in Bercovier–Pironneau [BP79] (see Fig. 6.1). The pressure is continuous in Ω and linear over each element $K_H \in \mathcal{T}_H$, while the velocity is

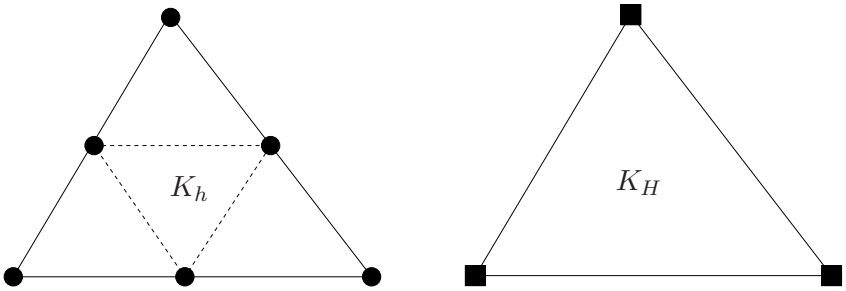


Figure 6.1: Velocity (left) and pressure (right) degrees of freedom for the ($\mathbb{P}_1$-iso$\mathbb{P}_2$)-$\mathbb{P}_1$ finite element discretization.

continuous in Ω and linear over each of the four subtriangles K_h, obtained by joining the midpoints of the edges of K. The union of all the subtriangles of the original triangulation defines a finer triangulation that will be referred to as $\mathcal{T}_h$.

This choice satisfies the inf-sup compatibility condition for the discrete problem [BF91] and provides the following linear convergence with respect to h for the Stokes problem:

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{\boldsymbol{V}} + \|p - p_h\|_Q \leq C\, h\, (\|\boldsymbol{u}\|_2 + \|p\|_1)\,. \tag{6.10}$$

The finite element formulation of the Navier–Stokes equations depends, therefore, on the following finite element spaces

$$\begin{aligned}
\boldsymbol{V}_h(\Omega) &= \left\{\boldsymbol{v}_h : \Omega \to \mathbb{R}^2,\, \boldsymbol{v}_h \in (\mathcal{C}^0(\bar{\Omega}))^2,\, \boldsymbol{v}_h|_{\partial\Omega} = 0,\, \boldsymbol{v}_h|_{K_h} \in P_1(K_h)^2,\, \forall K_h \in \mathcal{T}_h\right\},\\
Q_h(\Omega) &= \left\{q_h : \Omega \to \mathbb{R},\, q_h \in \mathcal{C}^0(\bar{\Omega}),\, q_h|_{K_H} \in P_1(K_H),\, \forall K_H \in \mathcal{T}_H\right\}.
\end{aligned}$$

For the approximation of the level-set equation in problem P6.1, we consider the same degrees of freedom as for the velocity. Therefore, the finite element space for the discrete level set function is

$$W_h(\Omega) = \left\{\psi_h : \Omega \to \mathbb{R},\, \psi_h \in \mathcal{C}^0(\Omega),\, \psi_h|_{K_h} \in P_1(K_h),\, \forall K_h \in \mathcal{T}_h\right\}.$$

The semi-discrete formulation of problem P6.1 reads:

P6.2 *Find $\boldsymbol{u}_h(t) \in \boldsymbol{V}_h$, $p_h(t) \in Q_h$ and $\phi_h(t) \in W_h$ such that:*

$$\begin{aligned}
m_{\phi_h}((\boldsymbol{u}_h)_t, \boldsymbol{v}_h) + a_{\phi_h}(\boldsymbol{u}_h, \boldsymbol{v}_h) + c_{\phi_h}(\boldsymbol{u}_h; \boldsymbol{u}_h, \boldsymbol{v}_h) &- b(\boldsymbol{v}_h, p_h)\\
&= m_{\phi_h}(\boldsymbol{g}, \boldsymbol{v}_h) + (\boldsymbol{f}_{\Gamma_h}, \boldsymbol{v}_h), \qquad \forall \boldsymbol{v}_h \in \boldsymbol{V}_h,\\
b(\boldsymbol{u}_h, q_h) &= 0, \qquad \forall q_h \in Q_h,\\
((\phi_h)_t, \psi_h) + l_{\boldsymbol{u}_h}(\phi_h, \psi_h) + j_h(\phi_h, \psi_h) &= 0, \qquad \forall \psi_h \in W_h,
\end{aligned}$$

for each $t \in]0, T]$.

In the level set equation we have added a stabilization term $j_h(\phi, \psi)$. Different choices for the stabilization of the level set equation can be adopted. In Chapter 7, we will introduce and analyse a new stabilization technique which has been found to be particularly suitable for this kind of problems.

6.2.1 Discretization of the surface tension term

The surface tension term in the continuous variational problem P6.1 reads

$$(\boldsymbol{f}_\Gamma, \boldsymbol{v}) = \int_\Omega \kappa\sigma\delta(\phi)\boldsymbol{n}\cdot\boldsymbol{v}\, d\boldsymbol{x} = \int_\Gamma \kappa\sigma\boldsymbol{n}\cdot\boldsymbol{v}\, d\boldsymbol{x}, \qquad \forall \boldsymbol{v}\in\boldsymbol{V}, \tag{6.11}$$

where we have used the fact that the action of δ on a smooth test function w is given by

$$\int_\Omega \delta_\Gamma(x)\, w(x)\, dx = \int_\Gamma v(s)\, ds. \tag{6.12}$$

One of the major advantages in the level-set method, when compared with other interface capturing methods (*e.g.* VOF) is the possibility of computing the interface normal $\boldsymbol{n}$ and curvature κ in a very convenient way. Indeed, if ϕ is the signed distance function from the interface Γ, normal (outgoing from Ω_2) and curvature are given, respectively, by

$$\boldsymbol{n} = \frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}, \tag{6.13}$$

$$\kappa = -\boldsymbol{\nabla}\cdot\boldsymbol{n} = -\boldsymbol{\nabla}\cdot\left(\frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}\right), \tag{6.14}$$

and they are defined in the entire domain Ω where the function ϕ is defined.

At the discrete level, the standard approach in the level set literature relies on the introduction of a smoothed Dirac function δ_ε, defined as:

$$\delta_\varepsilon(\phi(\mathbf{x})) = \left\{ \begin{array}{cc} \frac{1}{2\varepsilon}(1+\cos(\pi\phi(\mathbf{x})/\varepsilon), & \text{if } \phi(\mathbf{x})\le\varepsilon \\ 0, & \text{otherwise} \end{array} \right\}, \tag{6.15}$$

where ε is the smoothing parameter that prescribes the artificial thickness of the interface. Typically, the parameter ε is of the order of the grid size h. Given δ_ε, the surface tension term (6.11) can be computed as an integral term over Ω, as follows:

$$(\boldsymbol{f}_\Gamma, \boldsymbol{v}) = \int_\Omega \kappa\sigma\delta_\varepsilon(\phi)\boldsymbol{n}\cdot\boldsymbol{v} d\boldsymbol{x}, \qquad \forall \boldsymbol{v}\in\boldsymbol{V}. \tag{6.16}$$

The main drawback of this approach is that it gives rise to an error $\mathcal{O}(h)$ in the interface location.

In our scheme, the surface tension term is evaluated using the line integral in (6.11) along the interface Γ. The finite element approximation of (6.11) requires suitable approximations of $\boldsymbol{n}$ and κ at the interface. A piecewise continuous approximation $\boldsymbol{n}_h \in \boldsymbol{V}_h$ of the normal can be obtained solving the following variational problem:

$$\int_\Omega \boldsymbol{n}_h\cdot\boldsymbol{w}_h\, d\omega = \int_\Omega \frac{\boldsymbol{\nabla}\phi_h}{|\boldsymbol{\nabla}\phi_h|}\cdot\boldsymbol{w}_h\, d\omega, \qquad \forall \boldsymbol{w}_h\in\boldsymbol{V}_h. \tag{6.17}$$

It has been shown in [Smo01], that under suitable assumptions on the interface regularity, the following estimate holds for this approximation:

$$\|\boldsymbol{n}_h - \boldsymbol{n}\|_{L^\infty(\Gamma)} \le Ch^2\|\phi\|_{\mathcal{C}^3(\Omega)}, \tag{6.18}$$

with constant C independent of h.

We consider the following variational formulation for the curvature: find $\kappa \in L^2(\Omega)$ such that

$$\int_\Omega \kappa\psi \, d\boldsymbol{x} = \int_\Omega -\boldsymbol{\nabla}\cdot\boldsymbol{n}\psi \, d\boldsymbol{x} \tag{6.19}$$

$$= \int_\Omega \boldsymbol{n}\cdot\boldsymbol{\nabla}\psi \, d\boldsymbol{x} - \int_{\partial\Omega} \boldsymbol{n}\cdot\boldsymbol{n}_{\partial\Omega}\psi \, d\sigma, \qquad \forall\psi \in L^2(\Omega). \tag{6.20}$$

where $\boldsymbol{n}_{\partial\Omega}$ is the outgoing unit vector normal to $\boldsymbol{n}_{\partial\Omega}$.

A piecewise continuous approximation $\kappa_h \in W_h$ of the curvature can then be computed by solving the following discrete problem:

$$\int_\Omega \kappa_h\psi_h \, d\boldsymbol{x} = \int_\Omega \boldsymbol{n}_h\cdot\boldsymbol{\nabla}\psi_h \, d\boldsymbol{x} - \int_{\partial\Omega} \boldsymbol{n}_h\cdot\boldsymbol{n}_{\partial\Omega}\psi_h \, d\sigma, \qquad \forall\psi_h \in W_h. \tag{6.21}$$

The finite element approximation of the surface tension term (6.11) reads:

$$(\boldsymbol{f}_{\Gamma_h}, \boldsymbol{v}_h) = \int_{\Gamma_h} \kappa_h\sigma\boldsymbol{n}_h\cdot\boldsymbol{v}_h \, d\gamma \tag{6.22}$$

In the case of linear approximation of ϕ, the zero level set of ϕ_h is defined by a set on linear segments (see Fig. 6.2). Indeed, if we denote with $\{\boldsymbol{x}_i\}_{i=0,N_\Gamma}$ the set of intersection points between the zero level set of ϕ_h and the elements' edges, the discrete interface Γ_h is given by the union of the segments $\{\gamma_i = [\boldsymbol{x}_{i-1}, \boldsymbol{x}_i]\}_{i=1,N_\Gamma}$. The line integral (6.11) can then be computed elementwise over the interfacial line segments $\{\gamma_i\}_{i=1,\cdots,N_\Gamma}$:

$$\int_{\Gamma_h} \kappa_h\sigma\boldsymbol{n}_h\cdot\boldsymbol{v}_h \, d\gamma = \sum_i \int_{\gamma_i} \kappa_h\sigma\boldsymbol{n}_h\cdot\boldsymbol{v}_h \, d\gamma. \tag{6.23}$$

The integral over each interfacial segment can be evaluated using the standard 2-points Gaussian quadrature rules that is exact for polynomials of 3rd degree.

To assess the accuracy of the method adopted for the computation of the interface normal and curvature, we consider a simple test case where the interface is given by a circle of radius $r = 0.25$

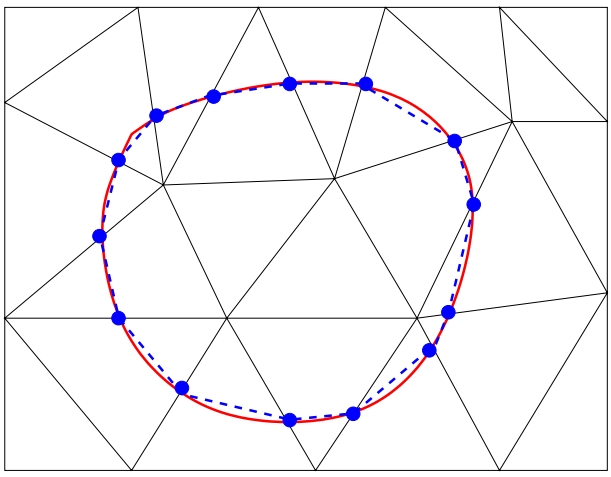


Figure 6.2: $\mathbb{P}_1$ approximation (dashed line) of the interface Γ.

with center in $(0.5, 0.5)$, In this case, the analytical expressions for normal and curvature are given by:

$$\boldsymbol{n} = \left(\frac{0.5 - x}{r}, \frac{0.5 - y}{r} \right),$$

$$\kappa = \frac{1}{r} = 4.$$

We consider the linear interpolant ϕ_I of the signed distance function ϕ from the circle. The discrete normal $\boldsymbol{n}_h$ (see (6.17)) and curvature κ_h (see 6.21) are computed for different mesh sizes and compared with the analytical values. The error is computed in the following discrete norms:

$$\begin{aligned}
\|\boldsymbol{n}_h - \boldsymbol{n}\|_{L^1} &= \frac{1}{N} \sum_{k=0}^{N_\Gamma} |\boldsymbol{n}_h(\boldsymbol{x}_k) - \boldsymbol{n}(\boldsymbol{x}_k)|, \\
\|\boldsymbol{n}_h - \boldsymbol{n}\|_{L^\infty} &= \max_{k=0,\ldots,N_\Gamma} |\boldsymbol{n}_h(\boldsymbol{x}_k) - \boldsymbol{n}(\boldsymbol{x}_k)|, \\
\|\kappa_h - \kappa\|_{L^1} &= \frac{1}{N} \sum_{k=0}^{N_\Gamma} |\kappa_h(\boldsymbol{x}_k) - \kappa(\boldsymbol{x}_k)|, \\
\|\kappa_h - \kappa\|_{L^\infty} &= \max_{k=0,\ldots,N_\Gamma} |\kappa_h(\boldsymbol{x}_k) - \kappa(\boldsymbol{x}_k)|.
\end{aligned}$$

The results presented in Tables 6.1 and 6.2 show second order convergence rate for both normal and curvature approximations.

h	L^∞	Order	L^1	Order
0.1	$6.72 \cdot 10^{-2}$		$2.54 \cdot 10^{-2}$	
0.05	$1.37 \cdot 10^{-2}$	2.21	$5.68 \cdot 10^{-3}$	2.11
0.025	$3.71 \cdot 10^{-3}$	1.92	$1.38 \cdot 10^{-3}$	2.02
0.0125	$9.86 \cdot 10^{-4}$	1.93	$3.51 \cdot 10^{-4}$	1.98
0.0625	$2.57 \cdot 10^{-4}$	1.95	$8.91 \cdot 10^{-5}$	1.98

Table 6.1: Numerical error on the normal.

h	L^∞	Order	L^1	Order
0.1	$7.30 \cdot 10^{-1}$		$1.90 \cdot 10^{-1}$	
0.05	$1.49 \cdot 10^{-1}$	2.21	$4.49 \cdot 10^{-2}$	2.05
0.025	$3.49 \cdot 10^{-2}$	2.06	$1.11 \cdot 10^{-2}$	2.01
0.0125	$8.47 \cdot 10^{-3}$	2.03	$2.75 \cdot 10^{-3}$	2.00
0.0625	$2.05 \cdot 10^{-3}$	2.03	$6.94 \cdot 10^{-4}$	1.99

Table 6.2: Numerical error on the curvature.

6.3 Time discretization and inexact factorization schemes

Denoting by $N_{\boldsymbol{u}}, N_p$ and N_ϕ the number of degrees of freedom of $\boldsymbol{u}_h, p_h$ and ϕ_h, respectively, problem P6.2 can be rewritten in algebraic form as follows:

P6.3 *Find $\boldsymbol{U}(t) \in \mathbb{R}^{N_{\boldsymbol{u}}}$, $\boldsymbol{P}(t) \in \mathbb{R}^{N_p}$ and $\Phi(t) \in \mathbb{R}^{N_\phi}$ such that*

$$\begin{aligned} M(\Phi(t))\boldsymbol{U}_t(t) + A(\Phi(t))\boldsymbol{U}(t) + C(\Phi(t),\boldsymbol{U}(t))\boldsymbol{U}(t) \;+\; & B^T\boldsymbol{P}(t) \\ = \; & \boldsymbol{G}(\Phi(t)) + \boldsymbol{F}_{\Gamma_h}(\Phi(t)), \\ B\boldsymbol{U}(t) \;=\; & 0, \\ H\Phi_t(t) + L(\boldsymbol{U}(t))\Phi(t) + J\Phi(t) \;=\; & 0. \end{aligned}$$

The matrices appearing in problem P6.3 are defined as follows:

$$\begin{aligned} M(\Phi(t))_{ij} &= m_{\phi_h}(\xi_i, \xi_j), \\ A(\Phi(t))_{ij} &= a_{\phi_h}(\xi_i, \xi_j), \\ C(\Phi(t),\boldsymbol{U}(t)))_{ij} &= c_{\phi_h}(\boldsymbol{u}_h; \xi_i, \xi_j), \\ B_{ij} &= b(\xi_i, \pi_j), \\ H_{ij} &= (\chi_i, \chi_j), \\ L(\boldsymbol{U})_{ij} &= l_{\boldsymbol{u}_h}(\chi_i, \chi_j), \\ J_{ij} &= j_h(\chi_i, \chi_j), \end{aligned}$$

where $\{\xi_i\}_{1\le i\le N_{\boldsymbol{u}}}$, $\{\pi_i\}_{1\le i\le N_p}$ and $\{\chi_i\}_{1\le i\le N_\phi}$ are the nodal basis functions of spaces $\boldsymbol{V}_h$, Q_h and W_h, respectively.

Different schemes can be adopted for the discretization of the time derivatives that appear in problem P6.3. In this section, we introduce the the first order implicit Euler scheme, although second-order schemes (*e.g.* Crank-Nicholson or Backward Difference Formula (BDF2)) can also be adopted straightforwardly. Nevertheless, the time error introduced by the splitting between Navier–Stokes and level set equations is of first order. The use of higher order time stepping would be justified only if a tighter coupling is adopted.

To reduce the computational complexity of the system, the solution of the Navier–Stokes equations is decoupled from that of the level-set equation. We consider a uniform decomposition of the time interval $[0, T]$ into N subintervals, namely, if $\Delta t = T/N$ is the time step, we use the index n to denote variables at time $t^n = n\,\Delta t$, with $n = 0, \ldots, N$. The time discretization of problem P6.3, combined with the decoupling of the Navier–Stokes equations from the level-set equation, leads to the solution, at each time step, of the algebraic Navier–Stokes system:

$$\begin{aligned} \frac{1}{\Delta t}\,M(\Phi^n)\boldsymbol{U}^{n+1} + A(\Phi^n)\boldsymbol{U}^{n+1} \;+\; & C(\Phi^n,\boldsymbol{U}^{n+1})\boldsymbol{U}^{n+1} + B^T P^{n+1} \\ = \; & \frac{1}{\Delta t}\,M(\Phi^n)\boldsymbol{U}^n + \boldsymbol{G}(\Phi^n) + \boldsymbol{F}_{\Gamma_h}(\Phi^n), \qquad (6.24) \\ B\boldsymbol{U}^{n+1} \;=\; & 0, \end{aligned}$$

followed by the solution of the algebraic level set system:

$$\frac{1}{\Delta t} H\Phi^{n+1} + L(\boldsymbol{U}^{n+1})\Phi^{n+1} + J\Phi^{n+1} = \frac{1}{\Delta t} H\Phi^{n}. \tag{6.25}$$

The nonlinearity in system (6.24) can be treated in different ways. Besides a *fully explicit strategy* (where $C(\Phi^n, \boldsymbol{U}^{n+1})\boldsymbol{U}^{n+1}$ is replaced by $C(\Phi^n, \boldsymbol{U}^n)\boldsymbol{U}^n$) and the classic *Newton linearization*, a typical approach is the *semi-implicit* one where $C(\Phi^n\boldsymbol{U}^{n+1})\boldsymbol{U}^{n+1}$ is replaced by the term $C(\Phi^n, \boldsymbol{U}^*)\boldsymbol{U}^{n+1})$, $\boldsymbol{U}^*$ being a suitable extrapolation of $\boldsymbol{U}^{n+1}$. In our scheme, we have set $\boldsymbol{U}^* = 2\boldsymbol{U}^n - \boldsymbol{U}^{n-1}$ (see e.g. [Gue99b]).

A common way to reduce the computational complexity associated to the solution of the Navier–Stokes system is based on the so-called *projection methods*. These methods typically compute the velocity and the pressure fields separately. An intermediate (non divergence-free) velocity is first computed, then the pressure is obtained solving a suitable Poisson-type problem and the final *end-of-step* velocity is computed enforcing the incompressibility constraint. Projection methods can be formulated either at the *differential* level (see, *e.g.,* [Cho68, GQ98]) or at the *algebraic* level (see, *e.g.*, [Per93, QSV00, QSV99]).

In the present work, we consider projection methods based on inexact algebraic factorization that are here briefly recalled. We consider the system (6.24) and we introduce the notations:

$$S(\Phi^n, \boldsymbol{U}^*) = \frac{1}{\Delta t} M(\Phi^n) + A(\Phi^n) + C(\Phi^n, \boldsymbol{U}^*),$$
$$\boldsymbol{F}(\Phi^n) = \left\{ \begin{array}{l} \dfrac{1}{\Delta t} M(\Phi^n)\boldsymbol{U}^n + \boldsymbol{G}(\Phi^n) + \boldsymbol{F}_{\Gamma_h}(\Phi^n) \\ 0 \end{array} \right\}.$$

Dropping the temporal index n and the explicit dependence of the matrices on Φ^n and $\boldsymbol{U}^*$ in order to simplify the notation, system (6.24) can be rewritten as:

$$\left[\begin{array}{cc} S & B^T \\ B & 0 \end{array} \right] \left\{ \begin{array}{c} \boldsymbol{U} \\ \boldsymbol{P} \end{array} \right\} = \left\{ \begin{array}{c} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \end{array} \right\}, \tag{6.26}$$

If we introduce the following block factorization:

$$\left[\begin{array}{cc} S & B^T \\ B & 0 \end{array} \right] = \left[\begin{array}{cc} S & 0 \\ B & -BS^{-1}B^T \end{array} \right] \left[\begin{array}{cc} I & S^{-1}B^T \\ 0 & I \end{array} \right], \tag{6.27}$$

system (6.26) can be solved exactly by the following three step algorithm:

$$\begin{array}{ll} (i) & S\widetilde{\boldsymbol{U}}^{n+1} = \boldsymbol{F}_1, \\ (ii) & -BS^{-1}B^T\boldsymbol{P}^{n+1} = \boldsymbol{F}_2 - B\widetilde{\boldsymbol{U}}^{n+1}, \\ (iii) & \boldsymbol{U}^{n+1} = \widetilde{\boldsymbol{U}}^{n+1} - S^{-1}B^T\boldsymbol{P}^{n+1}, \end{array} \tag{6.28}$$

or, equivalently, by its incremental form:

$$\begin{array}{ll} (i) & S\widetilde{\boldsymbol{U}}^{n+1} = \boldsymbol{F}_1 - B^T\boldsymbol{P}^{n}, \\ (ii) & -BS^{-1}B^T\delta\boldsymbol{P} = \boldsymbol{F}_2 - B\widetilde{\boldsymbol{U}}^{n+1}, \\ (iii) & \boldsymbol{U}^{n+1} = \widetilde{\boldsymbol{U}}^{n+1} - S^{-1}B^T\delta\boldsymbol{P}, \end{array} \tag{6.29}$$

where $\delta \boldsymbol{P} = \boldsymbol{P}^{n+1} - \boldsymbol{P}^n$.

Inexact factorization schemes are based on the solution of algorithms (6.28) or (6.29) with S^{-1} replaced by suitable approximations. Let $H^{(1)}$ and $H^{(2)}$ denote two approximations of S^{-1}. This choice correspond to the following inexact factorization

$$\begin{bmatrix} S & B^T \\ B & 0 \end{bmatrix} \approx \begin{bmatrix} S & 0 \\ B & -BH^{(1)}B^T \end{bmatrix} \begin{bmatrix} I & H^{(2)}\,B^T \\ 0 & I \end{bmatrix}, \tag{6.30}$$

We remark that the generalization of algebraic projection method to the case of variable coefficients (density and viscosity) is straightforward once the discrete system is written in the form (6.26).

6.3.1 Incremental Chorin–Temam projection scheme

A first possible choice of the two approximations of S^{-1} is the following:

$$H^{(1)} = \Delta t M_l^{-1}, \qquad H^{(2)} = \Delta t M_l^{-1},$$

where M_l is the lumped mass matrix [QV94]. With this choice algorithm (6.29) becomes:

$$\begin{aligned} &(i) \quad S\widetilde{\boldsymbol{U}}^{n+1} = \boldsymbol{F}_1 - B^T \boldsymbol{P}^n, \\ &(ii) \quad -\Delta t B M_l^{-1} B^T \delta \boldsymbol{P} = \boldsymbol{F}_2 - B\widetilde{\boldsymbol{U}}^{n+1}, \\ &(iii) \quad \boldsymbol{U}^{n+1} = \widetilde{\boldsymbol{U}}^{n+1} - \Delta t M_l^{-1} B^T \delta \boldsymbol{P}. \end{aligned} \tag{6.31}$$

It has been shown in [QSV00] that this choice yields a discrete scheme analogous to the one proposed in [Cho68] at the continuous level.

6.3.2 Incremental Yosida projection scheme

Another possible choice of the two approximations of S^{-1} is the following:

$$H^{(1)} = \Delta t M_l^{-1}, \qquad H^{(2)} = S^{-1}.$$

With this choice, algorithm (6.29) becomes:

$$\begin{aligned} &(i) \quad S\widetilde{\boldsymbol{U}}^{n+1} = \boldsymbol{F}_1 - B^T \boldsymbol{P}^n, \\ &(ii) \quad -\Delta t B M_l^{-1} B^T \delta \boldsymbol{P} = \boldsymbol{F}_2 - B\widetilde{\boldsymbol{U}}^{n+1}, \\ &(iii) \quad S\boldsymbol{U}^{n+1} = S\widetilde{\boldsymbol{U}}^{n+1} - B^T \delta \boldsymbol{P}. \end{aligned} \tag{6.32}$$

While in the Chorin–Temam projection scheme the momentum equation is perturbed by the inexact factorization, the Yosida projection scheme is based on a perturbation of the mass conservation equation. Indeed, the Yosida scheme can be interpreted as a *quasi-compressibility* scheme, with a penalizing term depending on the Yosida regularization of the Laplace operator (see [Bre83]).

The incremental Yosida projection scheme has been adopted in this work for the numerical simulations that will be described in Chapter 9. For the solution of the three sparse linear systems in (6.32), the Bi-Conjugate Gradient Stabilized (BiCGStab) iterative method preconditioned with an incomplete LU factorization has been used.

Remark. As already pointed out, the numerical scheme considered here is globally first-order in time due to the splitting between the Navier-Stokes and the level set equations. It is possible to construct projection methods based on inexact algebraic factorizations with higher order accuracy in time (see [SV]). A tighter coupling between the Navier–Stokes and the level set equations, in combination with these factorization schemes, could be used to obtain a numerical scheme for the solution of system (1.5)-(1.7) having higher accuracy in time. This improvement is still subject of investigations.

6.4 Overview of the solution algorithm

We have seen that an approximate solution to problem (1.5)-(1.7) can be obtained by sequentially solving, at each time step, the Navier–Stokes equations (with a given position of the interface) and a (stabilized) advection equation for the level set function. In Chapter 7, we will introduce and analyse a new stabilization technique for advection dominated problems.

The solution of free-surface problems by means of the level set method requires a further component, the so-called *reinitialization* step, which is a numerical procedure used to reconstruct the signed distance function from a given interface position. If a suitable redistancing algorithm is not adopted, during advection the level set function undergoes large deformations which can deteriorate the accuracy of the solution. Usually, it is not necessary to reinitialize the level set function at each time iteration. The reinitialization is performed every m time steps, with m depending on the specific problem (typically $m \approx 10$). A detailed discussion on the role of the reinitialization step, as well as the description and analysis of a new reinitialization approach, are given in Chapter 8.

A flowchart summarizing the algorithm introduced in this chapter is presented in Fig. 6.3.

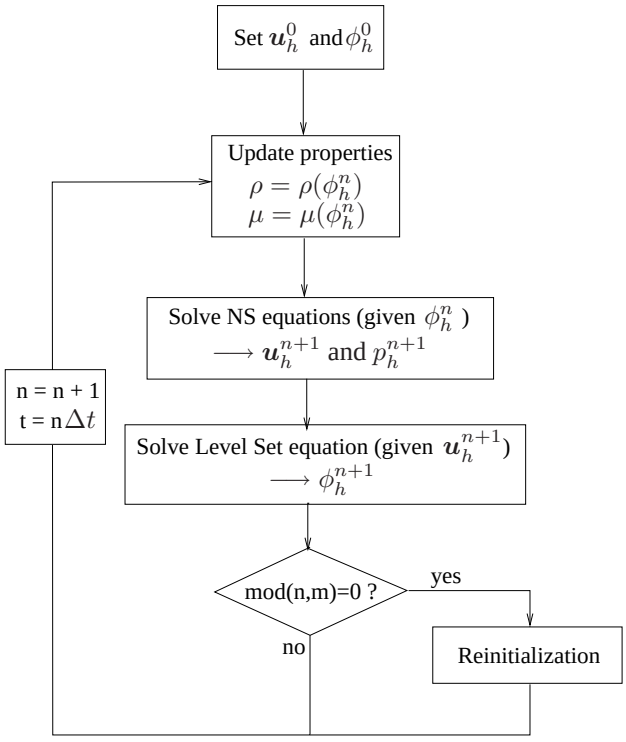


Figure 6.3: Overview of the level set algorithm.

Chapter 7

Subgrid Edge Stabilization for the Level Set equation

In Chapter 6, we have introduced a numerical method for the solution of viscous free-surface flow problems. The method is based on the Navier–Stokes equations coupled with the level set method, in which an time-dependent advection equation is solved to track to interface.

It is well known that the standard Galerkin method fails when the solution of advection dominated equations or pure advection equation is considered [Joh87, QV94]. The remedy consists either in resorting to discontinuous Galerkin methods [LR74], for which stability and local mass conservation is built into the method or using a stabilized Galerkin formulation for the continuous finite elements. The discontinuous Galerkin method takes the form of an interior penalty method whereas for the continuous case a Petrov-Galerkin approach is used. This dichotomy was questioned in a recent paper [Bur04] and it was shown that it is possible to construct interior penalty methods which are stable for the transport equation and of optimal order for both conforming and nonconforming finite element spaces. In the case of continuous approximation spaces a term must be added to the standard Galerkin formulation penalizing the jump of the gradient over internal element boundaries.

In the problem we are considering (the advection of the level set function), we want to represent an interface which is by definition continuous. Therefore, we use a continuous approximation to avoid extra difficulties due to a discontinuous approximated interface. In the framework of interior penalty methods for continuous approximation, we have complete freedom in the choice of time discretization and, as opposed to Petrov-Galerkin type stabilizations such as SUPG, we may use lumped mass and higher order time-stepping schemes of Runge-Kutta type.

Mass conservation is an important issue in the advection of the level set function. It is often argued that discontinuous finite element methods are locally conservative whereas continuous finite element methods are not. However in a recent paper [HEML00] it is shown that the mass flux for continuous finite element methods may be reconstructed solving an auxiliary local projection problem and that this flux is in fact conserved locally.

To obtain a numerical scheme with optimal mass conservation properties for continuous approximations, we propose to use the interior penalty term only on macro patches. We will show that such a discretization fits into the framework of the subgrid viscosity stabilization introduced by Guermond [Gue99a] (see also [BB04]) and yields a slightly more economic scheme. In fact thanks to the scale separation, the stabilization is only present on the finest scales and hence on the macro

scale the method will have the same conservation properties as the standard Galerkin method. This stabilization technique will be tested and compared with other existing methods with respect to time discretization and mass conservation.

The analysis carried out in this chapter has also been addressed in [BP04a].

7.1 A steady advection/reaction model problem

The stabilization technique that will be introduced and analysed in the following can be used for the solution of general advection dominated advection/diffusion/reaction problems. Since, in the framework of level-set method, we are dealing with the time dependent hyperbolic equation (6.1), the analysis will be presented for an advection/reaction problem. Moreover, for the sake of simplicity, we restrict our analysis to two dimensional problems.

Let Ω be an open bounded connected subset of $\mathbb{R}^2$ and $\boldsymbol{\beta}$ a smooth vector field such that $\boldsymbol{\nabla}\cdot\boldsymbol{\beta} = 0$. We consider in Ω the following general advection/reaction problem:

P7.1 *Find $\phi \in W^0_{\boldsymbol{\beta}}(\Omega)$ such that, for $f \in L^2(\Omega)$:*

$$\sigma\phi + \boldsymbol{\beta}\cdot\boldsymbol{\nabla}\phi = f$$

where σ is a positive constant.

The weak formulation of problem P7.1 reads:

P7.2 *Find $\phi \in W^0_{\boldsymbol{\beta}}(\Omega)$ such that*

$$a(\phi,\psi) = (f,\psi), \qquad \forall\psi \in L^2(\Omega),$$

where we have introduced the bilinear form:

$$a(\phi,\psi) = \int_\Omega \sigma\phi\psi d\boldsymbol{x} + \int_\Omega (\boldsymbol{\beta}\cdot\boldsymbol{\nabla}\phi)\psi d\boldsymbol{x}$$

The solution space and the test function space in problem P7.2 are different. We denote by W_h and L_h two finite dimensional approximation spaces of, respectively, $W^0_{\boldsymbol{\beta}}(\Omega)$ and $L^2(\Omega)$. From the standard approximation theory, we know that the couple (W_h, L_h) would yield optimal error bounds if $\dim(L_h) = \dim(W_h)$ and if the following discrete inf-sup condition is satisfied uniformly (with c independent of $\dim(L_h)$):

$$\exists c_s > 0, \forall\phi_h \in W_h, \qquad \sup_{\psi_h\in L_h} \frac{a(\phi_h,\psi_h)}{\|\psi_h\|_0} \geq c_s\|\phi_h\|_{1,\boldsymbol{\beta}}. \tag{7.1}$$

Standard Galerkin approximation corresponds to choose the same approximation space for solution and test functions ($W_h = L_h$). For this choice, condition (7.1) is not satisfied uniformly. A Petrov–Galerkin discretization of problem P7.2 should then involve two different approximation spaces (W_h, L_h) satisfying condition (7.1).

A possible alternative approach has been proposed by Guermond [Gue99a], where a generalized Galerkin approximation is considered. The method introduced in [Gue99a], which is referred to as *subgrid viscosity* stabilization, is based on a multiscale approach in which only the small scales are controlled by an artificial diffusion term. In the next Section, we recall the formulation of the subgrid viscosity stabilization and the relevant convergence results.

7.2 The subgrid viscosity stabilization

We consider a family of quasi-uniform triangulation $(\mathcal{T}_H)_H$, of the domain Ω, such that:

$$\exists \sigma \geq 1, \quad \max_{K_H \in \mathcal{T}_H} \frac{H_K}{\rho_K} \leq \sigma, \quad \forall H > 0, \tag{7.2}$$

$$\exists \tau > 0, \quad \min_{K_H \in \mathcal{T}_H} H_K \geq \tau H, \quad \forall H > 0, \tag{7.3}$$

where $H_K = \text{diam}(K_H)$, $\rho_K = \sup\{\text{diam}(S) \mid S \text{ is a ball contained in } K\}$ and the parameter

$$H = \max_{K_H \in \mathcal{T}_H} H_K$$

characterizes the mesh refinement.

From each triangle $K_H \in \mathcal{T}_H$, four triangles are created by connecting the midpoints of the edges. We set $h = H/2$ and denote by $K_h \in \mathcal{T}_h$ the resulting finer triangulation.

A two-level piecewise linear finite element approximation is defined by introducing the following two spaces:

$$X_H = \{\phi_H \in (W^0_{\boldsymbol{\beta}}(\Omega) \cap H^1(\Omega)) \mid \phi_{H|K_H} \in \mathbb{P}_1(K_H), \forall K_H \in \mathcal{T}_H\}, \tag{7.4}$$

$$X_h = \{\phi_h \in (W^0_{\boldsymbol{\beta}}(\Omega) \cap H^1(\Omega)) \mid \phi_{h|K_h} \in \mathbb{P}_1(K_h), \forall K_h \in \mathcal{T}_h\}. \tag{7.5}$$

Moreover, we introduce an additional discrete space $X_h^H \subset X_h$, such that we have the following decomposition:

$$X_h = X_H \oplus X_h^H. \tag{7.6}$$

Given $\phi_h \in X_h$ and $\phi_H \in X_H$ such that ϕ_h and ϕ_H coincide in the coarse scale nodes (see Fig. 7.1), the space decomposition (7.6) implies that $\phi_h^H = \phi_h - \phi_H, \forall \phi_h^H \in X_h^H$.

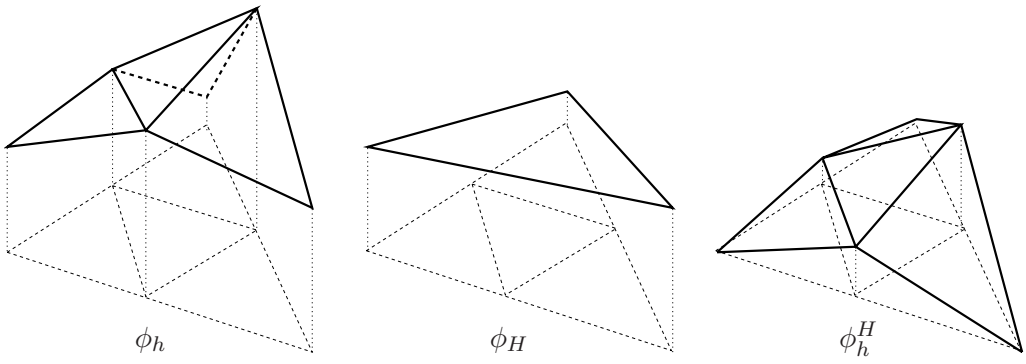


Figure 7.1: Subgrid decomposition: fine level solution ϕ_h (left), coarse level solution ϕ_H (center) and subgrid fluctuation $\phi_h^H = \phi_h - \phi_H$ (right).

The pair (X_H, X_h) will be referred to as the *two-level* $\mathbb{P}_1$ *setting*. Indeed, it shares the same structure of the $\mathbb{P}_1$-iso$\mathbb{P}_2$ finite element space introduced in Chapter 6 for the discretization of the Navier–Stokes equations.

This choice of spaces satisfies the following set of conditions (see [Gue99a] and references therein):

(i) Denoting with $P_H : X_h \to X_H$ the projection from X_h to X_H,

$$\exists c_s \geq 0, \forall \psi_h \in X_h, \quad \|P_H \psi_h\|_0 \leq c_s \|\psi_h\|_0.$$

(ii) The following inverse inequality holds in X_h

$$\exists c_i > 0, \forall \psi_h \in X_h, \qquad \|\psi_h\|_{0,\boldsymbol{\beta}} \leq c_i \, h^{-1} \|\psi_h\|_0. \tag{7.7}$$

(iii) The following interpolation property holds:

$$\exists c > 0, \forall \phi \in W^0_{\boldsymbol{\beta}}, \quad \inf_{\phi_H \in X_H} \{\|\phi - \phi_H\|_0 + H\|\phi - \phi_H\|_{1,\boldsymbol{\beta}}\} \leq cH^2 \|\phi\|_2. \tag{7.8}$$

(iv) The discrete inf-sup condition (7.1) holds for $W_h = X_H$ and $L_h = X_h$.

In the subgrid viscosity stabilization approach, the finite element discretization of problem P7.1, reads:

P7.3 *Find $\phi_h \in X_h$ such that*

$$a(\phi_h, \psi_h) + b_h(\phi_h^H, \psi_h^H) = (f, \psi_h), \qquad \forall \psi_h \in X_h.$$

where $b_h(\phi_h^H, \psi_h^H)$ is a stabilization term that satisfies the following additional condition

(v) There exists a norm $\|\cdot\|_{b_h}$ such that

$$\exists c_1, c_2 > 0, \forall \psi_h^H \in X_h^H, \quad c_1 \|\psi_h^H\|_{1,\boldsymbol{\beta}} \leq \|\psi_h^H\|_{b_h} \leq c_2 h^{-1} \|\psi_h^H\|_0, \tag{7.9}$$

and the bilinear form $b_h : X_h^H \times X_h^H \to \mathbb{R}$ satisfies the following coercivity and continuity properties: $\exists c_{c1}, c_{c2} > 0$, such that

$$b_h(\phi_h^H, \phi_h^H) \geq c_{c1} h \|\phi_h^H\|_{b_h}^2, \tag{7.10}$$

$$b_h(\phi_h^H, \psi_h^H) \leq c_{c2} h \|\phi_h^H\|_{b_h} \|\psi_h^H\|_b. \tag{7.11}$$

It has been proved in [Gue99a] (see also [EG04]) that, under conditions $(i) - (v)$, problem P7.3 has a unique solution and that, for a finite element approximation of degree $k = 1, 2$ the following *a priori* error estimates hold: if ϕ, the solution to problem P7.1, is in $H^{k+1}(\Omega)$, then the discrete solution ϕ_h to problem P7.3 satisfies

$$\|\phi - \phi_h\|_{1,\boldsymbol{\beta}} + \|\phi_h^H\|_b \leq ch^k \|\phi\|_{k+1}, \tag{7.12}$$

$$\|\phi - \phi_h\|_0 \leq ch^{k+1/2} \|\phi\|_{k+1}. \tag{7.13}$$

It is possible to construct different stabilization terms b_h which respect condition *(v)*. In [Gue99a], it has been shown that a subgrid viscosity stabilizing term, defined through the following bilinear form:

$$b_h(\phi_h^H, \psi_h^H) = s_h(\phi_h^H, \psi_h^H) := h(\boldsymbol{\nabla}\phi_h^H, \boldsymbol{\nabla}\psi_h^H), \tag{7.14}$$

together with the corresponding norm

$$\|\phi_h^H\|_{b_h} = \|\phi_h^H\|_{s_h} := \left(\int_\Omega (\boldsymbol{\nabla}\phi_h^H)^2 d\boldsymbol{x} \right)^{1/2}, \tag{7.15}$$

satisfies condition *(v)* and, therefore, Problem P7.3 has a unique solution.

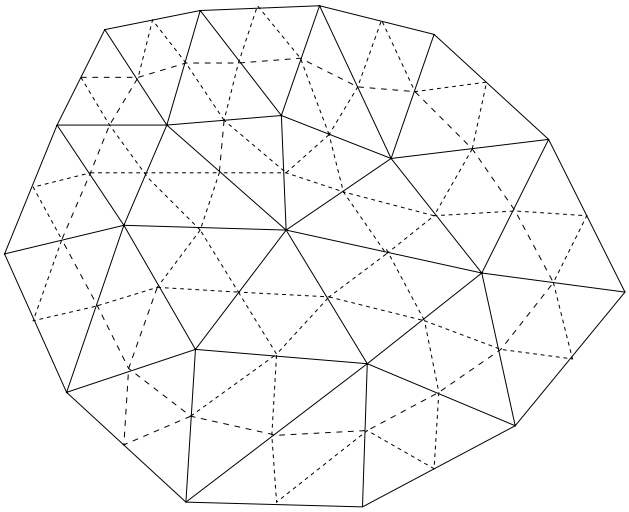


Figure 7.2: Subgrid internal edges (dashed line) where the edge stabilization term is added.

7.3 An edge-based subgrid viscosity stabilization

In [Bur04], it was shown that it is possible to construct interior penalty methods which are stable and of optimal order for advection/reaction/diffusion equations also for continuous approximation spaces. In this case, a term penalizing the jump of the gradient over all the internal edges of the triangulation must be added to the standard Galerkin approximation.

In the present work, we propose a local version of this interior penalty stabilization approach, that will be referred to as *subgrid edge stabilization*. In the framework of the subgrid viscosity stabilization, the term penalizing the jump of the gradient is added *only* over the edges of the subtriangles K_h internal to K_H (see Fig. 7.2), defined on each K_H as

$$E_i(K_H) = \{e_i \in \partial K_h \mid e_i \notin \partial K_H, \forall K_h \in K_H\}. \tag{7.16}$$

7.3.1 Two-level $\mathbb{P}_1$ interpolation

We introduce the following bilinear form

$$j_h(\phi_h, \psi_h) := \sum_{K_H} \sum_{e_i} \int_{e_i} h_{e_i}^2 [\boldsymbol{\nabla}\phi_h][\boldsymbol{\nabla}\psi_h]\, d\sigma, \tag{7.17}$$

and the associated norm:

$$\|\phi_h\|_{j_h} = \left(\sum_{K_H} \sum_{e_i} \int_{e_i} h_{e_i} [\boldsymbol{\nabla}\phi_h]^2 \, d\sigma \right)^{1/2}.$$

where $K_H \in \mathcal{T}_H$, $e_i \in E_i(K_H)$ and the quantity h_{e_i} denotes the size of edge e_i. Note that, in the case of two-level $\mathbb{P}_1$ interpolation we have:

$$\begin{aligned} j_h(\phi_h, \psi_h) &= j_h(\phi_h^H, \psi_h^H) \\ \|\phi_h\|_{j_h} &= \|\phi_h^H\|_{j_h} \end{aligned}$$

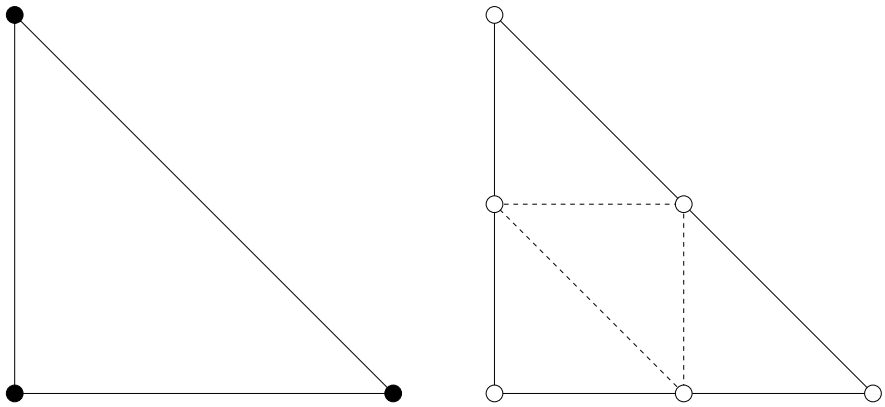


Figure 7.3: Reference two-level $\mathbb{P}_1$ element

since the jumps of ϕ_H and ψ_H across $e_i(K_H)$ are zero.

Properties $(i), (ii), (iii), (iv)$ in Section 7.2 only depend on the choice of the spaces X_H and X_h. As pointed out, they hold when a two-level $\mathbb{P}_1$ interpolation is considered. We can prove that, using the stabilization term (7.17), the solution to Problem P7.3 satisfies the error estimates (7.12) and (7.13). In order to do so, we have to prove that property (v) in Section 7.2 holds for $b_h(\cdot,\cdot) := j_h(\cdot,\cdot)$ and $\|\cdot\|_{b_h} := \|\cdot\|_{j_h}$. This is done in the following Lemma.

Lemma 7.3.1 *The norm $\|\cdot\|_{j_h}$ satisfies property (7.9) and the bilinear form $j_h(\cdot,\cdot)$ satisfies the coercivity and continuity inequalities (7.10) and (7.11).*

Proof To prove the first statement of the Lemma, it is enough to show that $\|\cdot\|_{j_h}$ is a norm in X_h^H equivalent to the standard subgrid viscosity norm $\|\cdot\|_{s_h}$.

We consider the reference $\mathbb{P}_1$-iso$\mathbb{P}_2$ element, where the right angle isosceles triangle $\widetilde{K}$ with vertices $(0,0), (1,0)$ and $(0,1)$ is split into four subtriangles $\widetilde{k}$. (see Fig. 7.3). For each $K_H \in \mathcal{T}_H$, we consider the affine mapping

$$F_{K_H}(\widetilde{\boldsymbol{x}}) = B_{K_H}\widetilde{\boldsymbol{x}} + \boldsymbol{b}_{K_H}, \qquad \forall \widetilde{\boldsymbol{x}} \in \widetilde{K}.$$

For each sub-triangle $\widetilde{k}$, we denote with ξ_1, ξ_2 and ξ_3 the three nodal basis $\mathbb{P}_1$ functions associated to the midpoints of the edges of $\widetilde{K}$. We introduce the following space

$$\widetilde{X}_h^H = \{\widetilde{\phi}_h^H \in \mathcal{C}^0(\widetilde{K}) \mid \widetilde{\phi}_h^H|_{\widetilde{k}} \in \text{span}\{\xi_1, \xi_2, \xi_3\}, \forall \widetilde{k} \in \widetilde{K}\},$$

and the norm

$$\|\widetilde{\phi}_h^H\|_{j_h,\widetilde{K}} = \left(\sum_{e_i \in E_i(\widetilde{K})} \int_{e_i} [\boldsymbol{\nabla}\widetilde{\phi}_h^H]^2 \, d\sigma \right)^{1/2}.$$

Clearly, $\|\widetilde{\phi}_h^H\|_{j_h,\widetilde{K}}$ satisfies the triangular inequality and $\forall \widetilde{\phi}_h^H \in \widetilde{X}_h^H$, $\|c\widetilde{\phi}_h^H\|_{j_h,\widetilde{K}} = |c|\|\widetilde{\phi}_h^H\|_{j_h,\widetilde{K}}$. To prove that $\|\cdot\|_{j_h,\widetilde{K}}$ is a norm on $\widetilde{X}_h^H$, we need to show that $\|\widetilde{\phi}_h^H\|_{j_h,\widetilde{K}} = 0$ if and only if $\widetilde{\phi}_h^H = 0$ on $\widetilde{K}$.

This is straightforward since, if

$$\sum_{e_i \in E_i(\widetilde{K})} \int_{e_i} [\boldsymbol{\nabla} \widetilde{\phi}_h^H]^2 d\sigma = 0$$

then $\boldsymbol{\nabla}\widetilde{\phi}_h^H$ is constant on $\widetilde{K}$. By definition $\widetilde{\phi}_h^H$ is null at the three vertices of $\widetilde{K}$, therefore $\widetilde{\phi}_h^H = 0$ on $\widetilde{K}$.

From the standard subgrid viscosity norm $\|\cdot\|_{s_h}$, we can derive the following norm on $\widetilde{X}_h^H$

$$\|\widetilde{\phi}_h^H\|_{s_h,\widetilde{K}} = \left(\int_{\widetilde{K}} (\boldsymbol{\nabla} \widetilde{\phi}_h^H)^2 d\boldsymbol{x} \right)^{1/2},$$

We introduce the notation $\preceq$ with $a \preceq b$ if and only if $a \leq c\, b$ where c is a positive constant independent of the grid.

By equivalence of norms on discrete spaces, using standard scaling estimates [GR86], on the generic triangle K_H we have:

$$\begin{aligned}
\|\phi_h^H\|_{s_h,K_H}^2 &= \det B_{K_H} \|\phi_h^H \circ F_{K_H}\|_{s_h,\widetilde{K}}^2 \\
&\preceq \det B_{K_H} \|\phi_h^H \circ F_{K_H}\|_{j_h,\widetilde{K}}^2 \\
&= \sum_{e_i \in E_i(\widetilde{K})} \int_{e_i} \frac{1}{\|B_{K_H}^{-T}\boldsymbol{n}\|} [\boldsymbol{\nabla}\phi_h^H \circ F_{K_H}]^2 \det B_{K_H} \|B_{K_H}^{-T}\boldsymbol{n}\| d\widetilde{s} \\
&\preceq \|B_{K_H}\| \sum_{e_i \in E_i(K_H)} \int_{e_i} [\boldsymbol{\nabla}\phi_h^H]^2 ds \qquad (7.18) \\
&\preceq h \sum_{e_i \in E_i(K_H)} \int_{e_i} [\boldsymbol{\nabla}\phi_h^H]^2 ds \\
&\preceq \|\phi_h^H\|_{j_h,K_H}^2 = \|\phi_h\|_{j_h,K_H}^2
\end{aligned}$$

where $\|\cdot\|_{s_h,K_H}$ and $\|\cdot\|_{j_h,K_H}$ are the restriction of $\|\cdot\|_{s_h}$ and $\|\cdot\|_{j_h}$ on the generic triangle K_H.

On the other hand, using the following trace inequality (see [Tho84])

$$\|\phi\|_{0,\partial K_h}^2 \preceq h^{-1}\|\phi\|_{0,K_h}^2 + h\|\phi\|_{1,K_h}^2, \qquad \forall \phi \in H^1(K_h), \qquad (7.19)$$

we have:

$$\begin{aligned}
\|\phi_h\|_{j_h,K_H}^2 &= \sum_{e_i \in E_i(K_H)} \int_{e_i} h_{e_i} [\boldsymbol{\nabla}\phi_h]^2 ds \\
&= \sum_{e_i \in E_i(K_H)} \int_{e_i} h_{e_i} [\boldsymbol{\nabla}(\phi_h - \phi_H)]^2 ds \\
&\leq \sum_{K_h \in K_H} \int_{\partial K_h} h \boldsymbol{\nabla}(\phi_h - \phi_H)^2 ds \qquad (7.20) \\
&\preceq \|\phi_h^H\|_{s_h,K_H}^2
\end{aligned}$$

Equations (7.18) and (7.20) summed over all $K_H \in \mathcal{T}_H$ give, respectively, the upper and lower bound for the norm equivalence:

$$\|\phi_h^H\|_{s_h} \preceq \|\phi_h\|_{j_h} \tag{7.21}$$

$$\|\phi_h^H\|_{j_h} \preceq \|\phi_h\|_{s_h} \tag{7.22}$$

We finally note that the coercivity inequality (7.10) is identically satisfied with $c = 1$ and continuity (7.11) is a direct consequence of Cauchy–Schwarz inequality.

■

Using the results of Lemma 7.3.1, we have that conditions $(i) - (v)$ in Section 7.2 are all satisfied for the proposed subgrid edge stabilization in the two-level $\mathbb{P}_1$ setting and error estimates (7.12) and (7.13) hold (with $k = 1$).

7.3.2 Two-level $\mathbb{P}_2$ interpolation

The subgrid edge stabilization introduced in the previous Section for linear finite element approximation can be extended to higher order approximations.

Let us consider the following the two-level $\mathbb{P}_2$ setting, where the spaces X_H and X_h are defined, in analogy with the linear case, as follows:

$$X_H = \{\phi_H \in (W_{\boldsymbol{\beta}}^0(\Omega) \cap H^1(\Omega)) \mid \forall K_H \in \mathcal{T}_H, \phi_{H|K_H} \in \mathbb{P}_2(K_H)\} \tag{7.23}$$

$$X_h = \{\phi_h \in (W_{\boldsymbol{\beta}}^0(\Omega) \cap H^1(\Omega)) \mid \forall K_h \in \mathcal{T}_h, \phi_{h|K_h} \in \mathbb{P}_2(K_h)\} \tag{7.24}$$

The degrees of freedom for the two space levels are shown in Fig. 7.4.

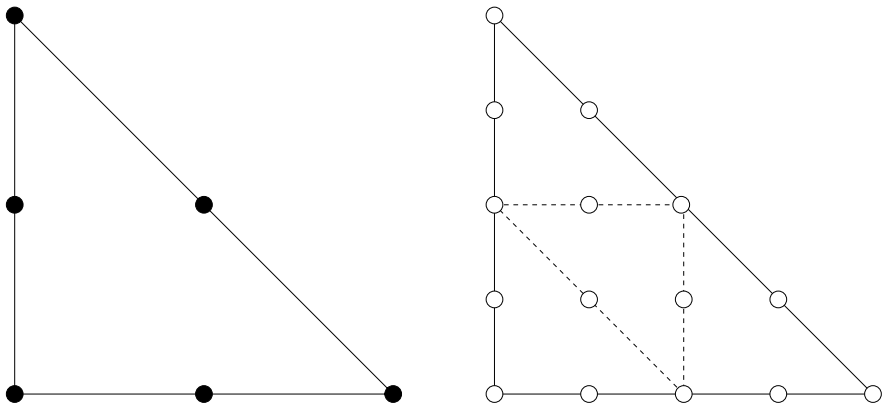


Figure 7.4: Reference two-level $\mathbb{P}_2$ element

In [Gue99a], it has been proved that, for the subgrid viscosity stabilization and this choice of finite element spaces, problem P7.3 has a unique solution and error estimates (7.12) and (7.13) hold (with $k = 2$)

We will show that it is possible to build a subgrid edge stabilization that enjoys the same convergence properties. In this case, the consistent stabilization term should penalize not only the jump of the

first derivatives, but also the jump of the second derivatives over the edges of the subtriangles K_h internal to K_H.

We consider the following bilinear form

$$j_h(\phi,\psi) := \sum_{K_H}\sum_{e_i}\int_{e_i}\left(h_{e_i}^2[\boldsymbol{\nabla}\phi][\boldsymbol{\nabla}\psi] + h_{e_i}^3[\boldsymbol{D}^2\phi][\boldsymbol{D}^2\psi]\right)\,d\sigma \tag{7.25}$$

and the associated norm:

$$\|\phi_h\|_{j_h} = (\|\phi_h\|_{j1_h}^2 + \|\phi_h\|_{j2_h}^2)^{1/2} \tag{7.26}$$

where

$$\|\phi_h\|_{j1_h} = \left(\sum_K\sum_{e_i}\int_{e_i} h_{e_i}[\boldsymbol{\nabla}\phi_h]^2\,d\sigma\right)^{1/2},$$

$$\|\phi_h\|_{j2_h} = \left(\sum_K\sum_{e_i}\int_{e_i} h_{e_i}^2[\boldsymbol{D}^2\phi]^2\,d\sigma\right)^{1/2},$$

and $\boldsymbol{D}^2(\cdot)$ denotes the Hessian operator.

The analysis is similar to the one carried out in Section 7.3.1 for the linear approximation. In particular, we need to show the equivalence of the two norms $\|\phi_h^H\|_{j_h}$ and $\|\phi_h^H\|_{s_h}$. This is done in the following Lemma.

Lemma 7.3.2 *The two norms $\|\cdot\|_{j_h}$ (defined in (7.26)) and $\|\cdot\|_{s_h}$ (defined in (7.15)) are equivalent.*

Proof We start considering the two-level $\mathbb{P}_2$ element on the reference triangle $\widetilde{K}$ split in the four sub-triangles $\widetilde{k}$ (see Fig. 7.4).

For each sub-triangle $\widetilde{k}$, we denote with ξ_1, ξ_2 and ξ_3 the three nodal basis $\mathbb{P}_2$ functions associated to the midpoints of the edges of $\widetilde{k}$. We introduce the following space

$$\widetilde{X}_h^H = \{\widetilde{\phi}_h^H \in \mathcal{C}^0(\widetilde{K}) \mid \widetilde{\phi}_h^H|_{\widetilde{k}} \in \text{span}\{\xi_1,\xi_2,\xi_3\}, \forall \widetilde{k} \in \widetilde{K}\},$$

and the norm

$$\|\widetilde{\phi}_h^H\|_{j_h,\widetilde{K}} = \left(\|\widetilde{\phi}_h^H\|_{j1_h,\widetilde{K}}^2 + \|\widetilde{\phi}_h^H\|_{j2_h,\widetilde{K}}^2\right)^{1/2},$$

where

$$\|\widetilde{\phi}_h^H\|_{j1_h,\widetilde{K}} = \left(\sum_{e_i\in E_i(\widetilde{K})}\int_{e_i}[\boldsymbol{\nabla}\widetilde{\phi}_h^H]^2\,d\sigma\right)^{1/2}.$$

$$\|\widetilde{\phi}_h^H\|_{j2_h,\widetilde{K}} = \left(\sum_{e_i\in E_i(\widetilde{K})}\int_{e_i}[\boldsymbol{D}^2\widetilde{\phi}_h^H]^2\,d\sigma\right)^{1/2}.$$

We can show that $\|\widetilde{\phi}_h^H\|_{j_h,\widetilde{K}} = 0$ if and only if $\widetilde{\phi}_h^H = 0$ on $\widetilde{K}$. In fact, if

$$\sum_{e_i\in E_i(\widetilde{K})}\int_{e_i}\left([\boldsymbol{\nabla}\widetilde{\phi}_h^H]^2 + [\boldsymbol{D}^2\widetilde{\phi}_h^H]^2\right)d\sigma = 0$$

then $\widetilde{\phi}_h^H \in \mathcal{C}^2(\widetilde{K})$. Moreover, by definition of the space $\widetilde{X}_h^H$, $\widetilde{\phi}_h^H \in \mathbb{P}_2$ on each of the 4 subtriangles $\widetilde{k}$, therefore $\widetilde{\phi}_h^H \in \mathbb{P}_2(\widetilde{K})$. Since $\widetilde{\phi}_h^H$ is null at the six vertices of the four subtriangles $\widetilde{k}$, we conclude that $\widetilde{\phi}_h^H = 0$ on $\widetilde{K}$. This implies that $\|\cdot\|_{j_h,\widetilde{K}}$ is a norm on $\widetilde{X}_h^H$.

As in (7.20), we can use equivalence of norms on discrete spaces and standard scaling estimates and we get:

$$\begin{aligned}
\|\phi_h^H\|_{s_h,K_H}^2 &= \det B_{K_H} \|\phi_h^H \circ F_{K_H}\|_{s_h,\widetilde{K}}^2 \\
&\preceq \det B_{K_H} \|\phi_h^H \circ F_{K_H}\|_{j1_h,\widetilde{K}}^2 + (\det B_{K_H})^2 \|\phi_h^H \circ F_{K_H}\|_{j2_h,\widetilde{K}}^2 \\
&\preceq \sum_{e_i(K_H)} \int_{e_i} h_{e_i} [\boldsymbol{\nabla}\phi_h^H]^2 ds + \sum_{e_i(K_H)} \int_{e_i} h_{e_i}^2 [\boldsymbol{D}^2 \phi_h^H]^2 ds \qquad (7.27) \\
&\preceq \|\phi_h^H\|_{j_h,K_H}^2 = \|\phi_h\|_{j_h,K_H}^2
\end{aligned}$$

The lower bound

$$\|\phi_h^H\|_{j_h,K_H} \preceq \|\phi_h^H\|_{s_h,K_H} \qquad (7.28)$$

is again a direct consequence of trace inequality (7.19) and inverse inequality.

The upper and lower bounds for the norm equivalence are obtained summing, respectively, equations (7.27) and (7.28) over all $K_H \in \mathcal{T}_H$:

$$\|\phi_h^H\|_{s_h} \preceq \|\phi_h\|_{j_h} \qquad (7.29)$$

$$\|\phi_h^H\|_{j_h} \preceq \|\phi_h\|_{s_h} \qquad (7.30)$$

■

Remark. In practice, for both subgrid viscosity and subgrid edge methods, the stabilization term has to be weighted with a stabilization parameter (independent of h) which should be chosen proportional to $|\boldsymbol{\beta}|$ for dimensional reasons.

In conclusion, we have introduced and analysed a new local version of the edge stabilization technique and we have shown that it can be interpreted in the framework of the subgrid viscosity stabilization, from which it inherits the (quasi)-optimal convergence properties for linear and quadratic finite element approximations. The subgrid edge stabilization has been employed in the solution of the level set step of the algorithm described in Section 6.4. In this respect, the two-level $\mathbb{P}_1$ setting has been adopted, making use of the two-level grid and data structures employed for the ($\mathbb{P}_1$-iso$\mathbb{P}_2$)-$\mathbb{P}_1$ finite element discretization of Navier–Stokes system. The technique introduced in this chapter can be extended to the stabilization of the Navier–Stokes equations. A similar approach was proposed in [Sil94] for the stabilization of the Stokes equations using $\mathbb{P}_1$-$\mathbb{P}_1$ finite elements.

7.4 Numerical examples

In this section, we present some numerical examples in order to assess the theoretical results obtained above and to highlight the advantages of the proposed stabilization method when compared with more standard stabilization approaches.

7.4.1 Advection-reaction test case

We consider the following advection-reaction problem

$$\sigma\phi + \beta \cdot \nabla\phi = f \quad \text{in } \Omega,$$

with $\sigma = 1$, $\beta = (1, 0)$, $\Omega = [0, 1]^2$ with two different source terms f corresponding to the following analytical solutions:

- Test case 1: $\phi = \exp\left(-\dfrac{(x-0.5)^2}{0.2} - \dfrac{3(y-0.5)^2}{0.2}\right)$
- Test case 2: $\phi = \dfrac{1}{2}\left(1 - \tanh\left(\dfrac{x-0.5}{0.05}\right)\right)$

The two exact solutions considered are displayed in Fig. 7.5. We have computed the solution using the subgrid edge stabilization method and the standard Streamline Upwind Petrov Galerkin method (see *e.g.* [BH82]). The stabilization parameter for the subgrid edge stabilization was chosen to be $\gamma_E = 0.025$, while for SUPG we have used $\delta_{SUPG} = h_K/(\sqrt{15}|\boldsymbol{\beta}|)$, a value which has been shown to be optimal for advection-reaction equation (see [HB82]). The spatial convergence tests for the two test case have been performed over four uniform unstructured grids with $h = 1/20, 1/40, 1/80, 1/160$ and the L^2, H^1 and L^∞ errors have been computed. The results are presented in Tables 7.1 and 7.2 together with the corresponding convergence order.

For both subgrid edge and SUPG stabilization, we obtain second order convergence in L^2 and L^∞ norms and first order convergence in H^1 norm, conforming to the theoretical predictions. Indeed, it has been pointed out with different stabilization methods for advection/diffusion/reaction problems (see [Bur04] for edge stabilization and [Zho97] for streamline diffusion) that the convergence error in the L^2-norm rarely degenerate to $\mathcal{O}(h^{3/2})$, unless special computational grids (*e.g.* Peterson grid) are considered.

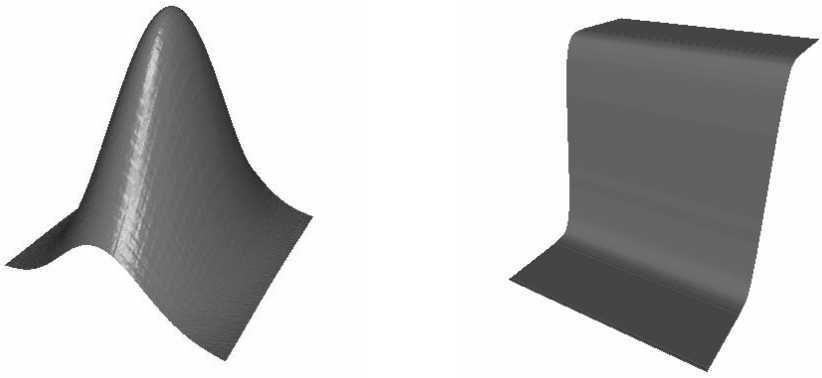


Figure 7.5: Gaussian (left) and hyperbolic (right) tangent exact solutions for the advection/reaction test case.

7.4.2 Linear advection problem

We consider a second test case with the aim of assessing the mass conservation properties of the proposed stabilization technique when dealing with time dependent hyperbolic problems.

We consider the following pure advection problem:

	Subgrid Edge						SUPG					
h	L^2		H^1		L^∞		L^2		H^1		L^∞	
0.05	6.7e-3	(1.9)	4.1e-1	(0.7)	5.8e-2	(1.5)	4.6e-3	(2.0)	3.1e-1	(1.0)	5.5e-2	(1.0)
0.025	2.0e-3	(2.0)	2.6e-1	(0.9)	2.0e-2	(1.8)	1.2e-3	(2.1)	1.6e-1	(1.0)	1.5e-2	(2.0)
0.0125	5.1e-4	(2.0)	1.4e-1	(1.0)	5.9e-3	(2.1)	2.7e-4	(2.0)	7.9e-2	(1.0)	3.6e-3	(2.3)
0.00625	1.3e-4		7.2e-2		1.4e-3		7.1e-5		3.9e-2		7.5e-4	

Table 7.1: Convergence results for the advection/reaction test 1 (gaussian).

	Subgrid Edge						SUPG					
h	L^2		H^1		L^∞		L^2		H^1		L^∞	
0.05	7.4e-3	(2.1)	5.5e-1	(1.1)	7.6e-2	(2.0)	7.0e-3	(2.2)	5.1e-1	(1.2)	5.5e-2	(1.7)
0.025	1.8e-3	(1.9)	2.6e-1	(0.9)	1.9e-2	(1.8)	1.6e-3	(2.1)	2.2e-1	(1.1)	1.6e-2	(2.1)
0.0125	4.8e-4	(2.0)	1.4e-1	(1.0)	5.6e-3	(2.0)	3.7e-4	(2.0)	1.1e-1	(1.0)	3.9e-3	(2.0)
0.00625	1.2e-4		7.8e-2		1.7e-3		9.1e-5		5.1e-2		9.8e-4	

Table 7.2: Convergence results for the advection/reaction test 2 (hyperbolic tangent).

P7.4 *Given the velocity field $\boldsymbol{\beta} = (1, 0)$, for each $t \in]0, T]$, find $\phi(x, t) \in W_{\boldsymbol{\beta}}$ such that*

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \phi = 0$$

where Ω is the unit square and a Dirichlet boundary condition is imposed at the inflow. The initial value of ϕ is the signed distance function from the circle center in (0.3,0.5) and radius R=0.15 (see Fig. 7.6).

The SUPG stabilization for time-dependent hyperbolic problems (such as P7.4) consists in adding to the standard Galerkin approximation of P7.4, the following term:

$$a_{SUPG}(\phi_h, \psi_h) = \sum_{k \in \mathcal{T}_h} \delta_{SUPG} \int_k \left(\frac{\partial \phi_h}{\partial t} + \boldsymbol{\beta} \cdot \boldsymbol{\nabla} \phi_h \right) \left(\frac{\partial \psi_h}{\partial t} + \boldsymbol{\beta} \cdot \boldsymbol{\nabla} \psi_h \right) d\boldsymbol{x}. \tag{7.31}$$

We note that a full consistent implementation of the SUPG stabilization involves a time derivative term also in the test function. This would require a space-time finite element discretization, which is more complex to implement and requires additional unknowns. Therefore, we consider a non-consistent streamline upwind (SU) stabilization introducing the stabilization term

$$a_{SU}(\phi_h, \psi_h) = \sum_{k \in \mathcal{T}_h} \delta_{SU} \int_k (\boldsymbol{\beta} \cdot \boldsymbol{\nabla} \phi_h)(\boldsymbol{\beta} \cdot \boldsymbol{\nabla} \psi_h) d\boldsymbol{x}. \tag{7.32}$$

For the subgrid edge stabilization, a rigorous analysis on the choice of the stabilization parameter (that we denote δ_E) is still missing. In the present case, we have considered $\delta_E = 0.025$, while for the SU stabilization, we consider the same value of the stabilization parameter used for the advection/reaction test case $\delta_{SU} = h_K/(\sqrt{15}|\boldsymbol{\beta}|)$.

We consider two different finite difference time discretization schemes: a first order implicit Euler scheme

$$\frac{\partial \phi_h}{\partial t}(t^{k+1}) \approx \frac{\phi_h^{k+1} - \phi_h^k}{\Delta t}$$

and a second order Backward Difference Formula (BDF2):

$$\frac{\partial \phi_h}{\partial t}(t^{k+1}) \approx \frac{3\phi_h^{k+1} - 4\phi_h^k + \phi_h^{k-1}}{2\Delta t}$$

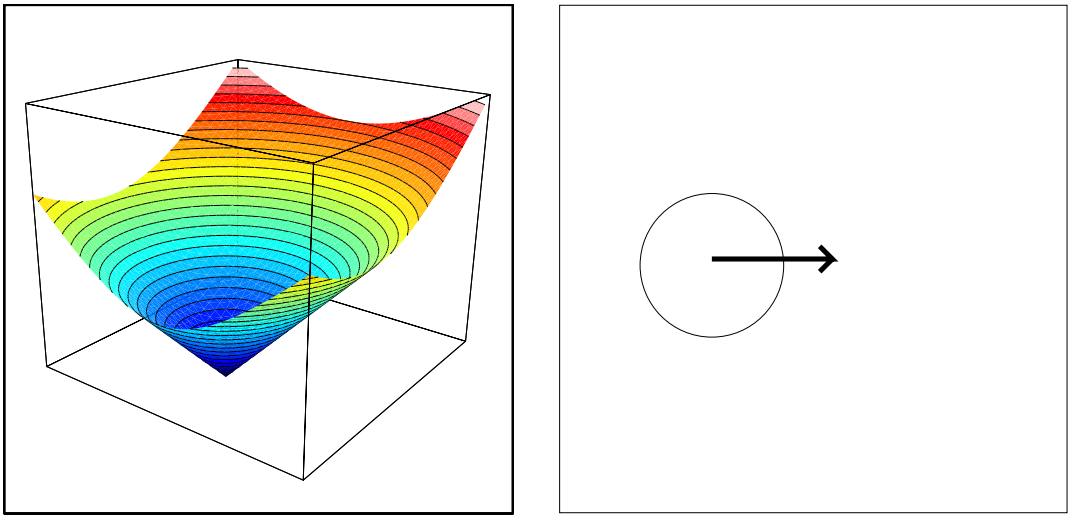


Figure 7.6: Linear advection test case: the initial level set function (left) and and its zero level set (right).

In Fig. 7.7, we present the time evolution of the area inside the zero level set of ϕ_h computed using the two stabilization techniques and first order and second order time-stepping schemes. Two different time steps, $\Delta t = 0.01$ and $\Delta t = 0.001$, have been considered for each case.

These results show the better mass conservation of the proposed edge stabilization technique with respect to the SU stabilization, especially when high order schemes in time are considered. We can observe in Fig. 7.7 the effect of the non-consistency in the SU formulation: when the time step is reduced, the first order Euler scheme and the second order BDF2 scheme behave almost identically in term of mass conservation. The zero level sets at different time instants obtained using subgrid edge and SU stabilizations are compared in Fig. 7.8 (with BDF2 scheme and $\Delta t = 0.001$).

Although, at the present stage, an optimal choice of the stabilization parameter δ_E is not available, it is interesting to note that the mass conservation of the subgrid edge stabilization is almost

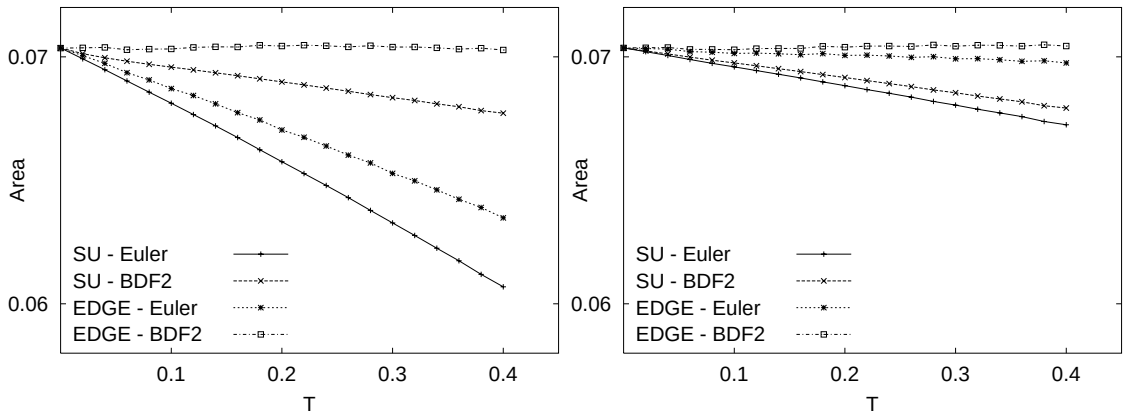


Figure 7.7: Global mass conservation for the linear advection test case with different stabilizations and time discretization: time evolution of the area inside the zero level set on a grid with $h = 1/40$ and time step $\Delta t = 0.01$ (left) and $\Delta t = 0.001$ (right).

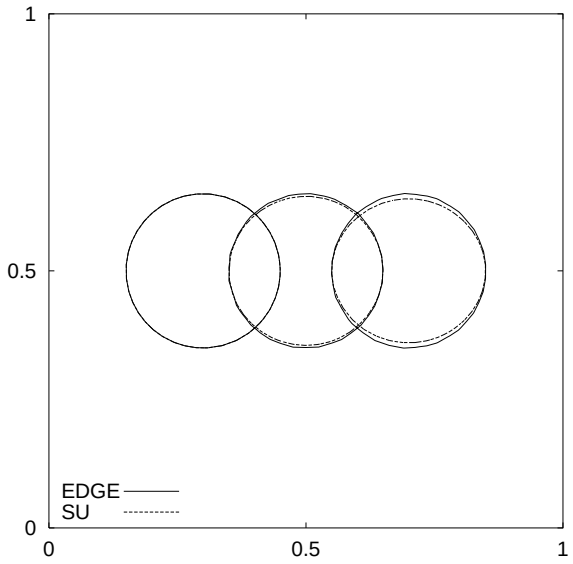


Figure 7.8: Zero level sets for the linear advection test case at time instants t=0, 0.2, 0.4 s obtained using subgrid edge and SU stabilizations (with BDF2 scheme and $\Delta t = 0.001$).

independent of the value of δ_E. This is not the case when the SU stabilization is considered. In Fig. 7.9, we present the time evolution of the area inside the zero level set for the two stabilization techniques and different values of the stabilization parameters. Indeed, as observed above, given the local nature of the proposed stabilization technique, the stabilization is only present on the finest scales and hence on the macro scale the method will have the same conservation properties as the standard Galerkin method.

Remark. We would like to observe that better results in term of mass conservation can be obtained if the consistent SUPG stabilization term (7.31) is used in the framework of space-time finite element approximation. Another option that could enhance mass conservation can be obtained neglecting the time derivative term in (7.31) in the test function. This would allow to use a finite difference method for time discretization.

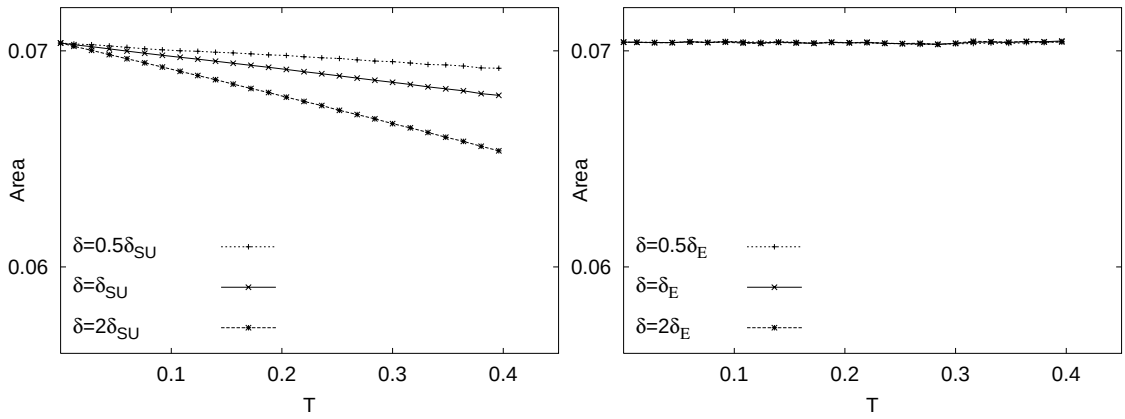


Figure 7.9: Dependence of the mass conservation on the stabilization parameters for SU (left) and subgrid edge (right) stabilizations.

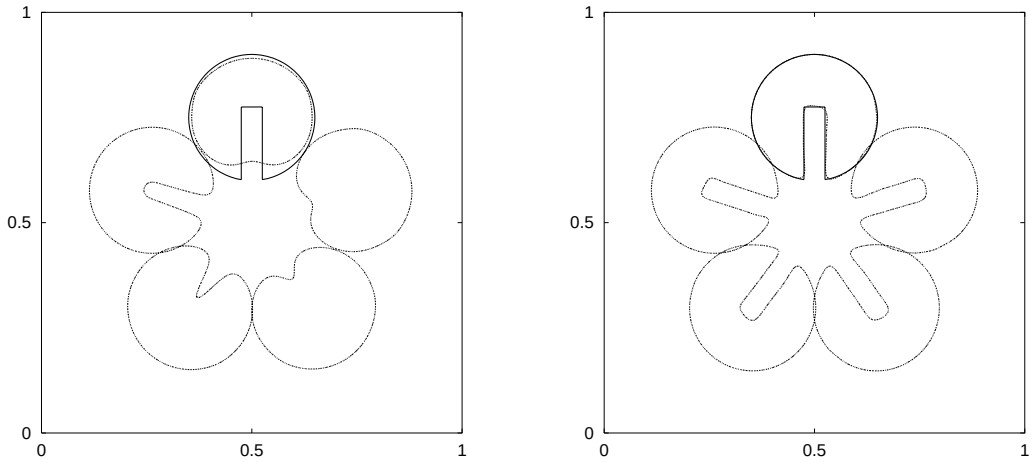


Figure 7.10: Zalesak slotted disk test case (rigid rotation): SU stabilization (left) and subgrid edge stabilization (right), $h = 1/160$. Solid line represents the exact solution.

7.4.3 Solid body rotation

The last test case considered in this chapter is the so-called Zalesak problem [Rud97] in which a slotted circle is rigidly rotated through one or more revolutions. This problem has been widely used in the literature as a test for numerical schemes for scalar advection problems.

The computational domain is the unit square $\Omega = [0, 1] \times [0, 1]$ and the slotted circle is rigidly advected by a given velocity field $\boldsymbol{u} = (0.5 - y, -0.5 + x)$. The initial value of ϕ is the signed distance function from the slotted circle. The circle is initially centered at $(0.5, 0.75)$ and has radius R=3/20. The slot width is 1/20. The problem has been solved using subgrid edge and SU stabilization on two uniform unstructured grids with $h = 1/80$ and $h = 1/160$, covering one rotation with 1000 time steps. The same stabilization parameters as in the linear advection test case have been used. The interface locations at different time instants computed with the two stabilizations are shown in Fig. 7.10. These results confirm the better behaviour of the proposed method. A detail of the interface position after one rotation obtained with the two stabilization on grid with increasing refinement is displayed in 7.11.

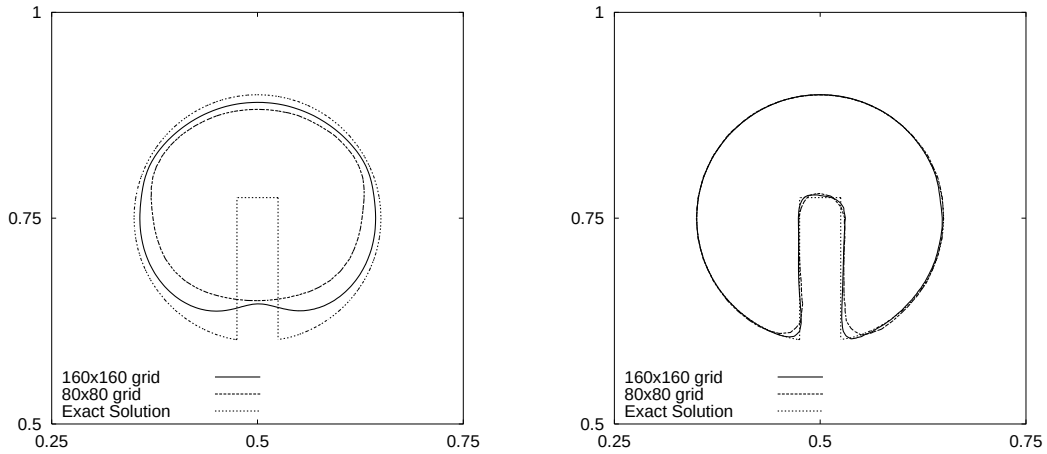


Figure 7.11: Zalesak slotted disk after one revolution: SU stabilization (left) and subgrid edge stabilization (right), $h = 1/80$ and $h = 1/160$.

Chapter 8

The Level Set Reinitialization

Most of the advantages associated with the level set approach, when compared to other front capturing techniques for free-surface problems (*e.g.* Volume of Fluid), rely on the fact that the transported level set function is smooth at the interface. In particular, we have seen in Chapter 6 that the level set is initialized as the signed distance function from the interface.

Unfortunately, the property of the level set function being a distance function is not preserved during advection. As a matter of fact, regions where the level set function becomes *too flat* are generated during advection, with a loss of accuracy in the determination of the interface location. On the other hand, when the level set function becomes *too steep*, a nearly discontinuous function has to be advected, with the need of setting up appropriate discontinuity capturing schemes to reduce numerical oscillation near the interface.

To prevent the deterioration of the level-set function, a suitable *reinitialization* procedure should be introduced. Consider the function $\widetilde{\phi} \in W_h$, not a distance function, whose zero level set $\Gamma = \{\boldsymbol{x} \in \Omega \mid \widetilde{\phi}(\boldsymbol{x}) = 0\}$ represents the interface at a given time. The *reinitialization* consists in constructing a new function $\phi \in W_h$, with the property that its zero level set is the same as the one of ϕ_0 and such that ϕ is the signed distance function from the interface (see Fig. 8.1).

The need of a suitable reinitialization step in the solution of complex free-surface problems was first pointed out in [DT80]. The idea has then been reconsidered in [SSO94] in the context of the level-set method.

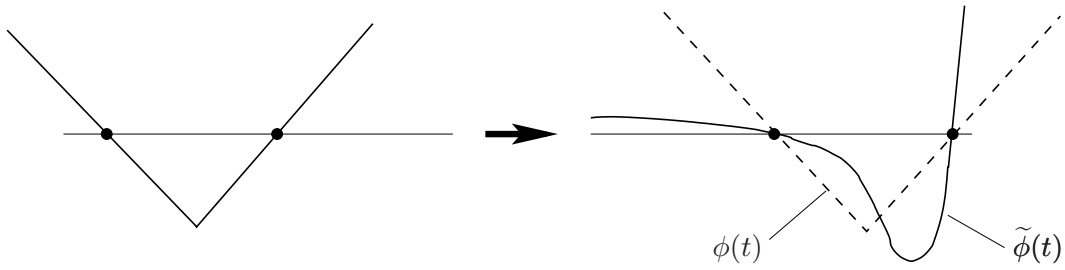


Figure 8.1: 1D sketch of the reinitialization procedure: an initial distance function (left) is advected and, at time t, $\widetilde{\phi}(t)$ is replaced by a reinitialized signed distance function $\phi(t)$ (right, dashed line) from the new interface position.

Several different reinitialization procedures have been proposed in literature, see *e.g.* [SSO94, CS02, PMO+99, CHMO96, SF99]. In this chapter, we will discuss some possible reinitialization approaches and we will introduce a new technique.

The analysis carried out in this chapter has also been addressed in [BP04b].

8.1 Standard reinitialization procedures

The different reinitialization techniques proposed in the level set literature can be classified in two main categories: direct methods (where the interface is explicitly localized and the distance from the interface for each computational node is computed) and PDE-based method (where the reinitialized level set function is obtained as solution of an additional PDE equation).

Moreover, for most reinitialization techniques found in the literature, local versions have been proposed (see *e.g.* [PMO+99, Smo01]). In fact, since we are interested in an accurate solution of the level set function close to the interface, the reinitialization can be performed only on a layer around the interface, in order to reduce the computational complexity.

8.1.1 Direct reinitialization

In the direct approach (see, *e.g.*, [Smo01]), the interface location is computed explicitly. In the case of a $\mathbb{P}_1$ approximation, this corresponds to the computation of the set of linear segments $\{\gamma_i\}_{i=1,\cdots,N_\Gamma}$, where $\widetilde{\phi}_h \in W_h$, solution of equation (6.25) at a given time, is zero (see Sec. 6.2.1 and Fig. 6.2). For each node $\boldsymbol{x}_k$ of the computational grid, the distance is then computed directly by:

$$\phi_h(\boldsymbol{x}_k) = \min_{i=1,\cdots,N_\Gamma} (\text{dist}(\boldsymbol{x}_k, \gamma_i)). \tag{8.1}$$

The reinitialized level set function $\phi_h \in W_h$ is then defined as the piecewise linear interpolation of the node values computed in (8.1). The main advantage of this technique is its robustness. Moreover, it is very easy to implement. On the other hand, if an appropriate local technique is not adopted, this approach turns out to be excessively expensive, in particular when three-dimensional problems are considered. Furthermore, this technique has been found to suffer more in terms of mass conservation than other approaches, as it will be pointed out in Section 8.3.

8.1.2 PDE-based reinitialization

In the work by Sussman *et al.* [SSO94], a reinitialization technique based on the solution of an additional PDE equation was first proposed.

Given $\widetilde{\phi}_h$, solution of the level set advection equation (6.25), the reinitialized level set function ϕ_h is obtained as being the steady state of the solution of the following Hamilton–Jacobi equation:

$$\begin{cases} \dfrac{\partial \phi_h}{\partial \tau} = \text{sign}(\widetilde{\phi}_h)(1 - |\boldsymbol{\nabla}\phi_h|), & \tau \geq 0, \\ \phi_{h|\tau=0} = \widetilde{\phi}_h, \end{cases} \tag{8.2}$$

where "sign" denotes the signum function and τ is a *pseudo*-time. This method does not require to explicitly localize the interface. On the other hand, the approximation of equation (8.2) demands suitable numerical schemes (see, *e.g.*, [CL83, OS88, OS91, CFN95, FF02]) which become particularly complex when combined with unstructured computational grid (see, *e.g.*, [BS98]). Moreover, any numerical scheme for the solution of equation (8.2) introduces a certain amount of numerical diffusion. In particular, it has been shown in [SSO94] that a suitable smoothing of the signum function in equation (8.2) is required for numerical stability reasons. This approximation results in an artificial displacement of the interface position and downgrades the global mass conservation properties of the scheme (see, *e.g.*, [Tor00]).

8.2 Interface local projection reinitialization

In this Section, we will present a new technique, which will be referred to as *interface local projection reinitialization*, with the aim of fulfilling the requirements that an optimal reinitialization should satisfy, namely accuracy at the interface and efficiency. With respect to the reinitialization step, it is convenient to split the computational domain in two regions and treat them separately (see Fig 8.2). These two regions are

- an **interface region**, defined as $\Omega_{\text{int}} = \{\bigcup K_h : K_h \in \mathcal{T}_h, K_h \cap \Gamma \neq \emptyset\}$, union of the elements intersected by the interface, where the highest accuracy is needed to avoid artificial interface displacement;
- a **"far-field" region**, defined as $\Omega_{\text{out}} = \Omega \setminus \Omega_{\text{int}}$, away from the interface, where efficient reinitialization procedures are needed and accuracy is less critical.

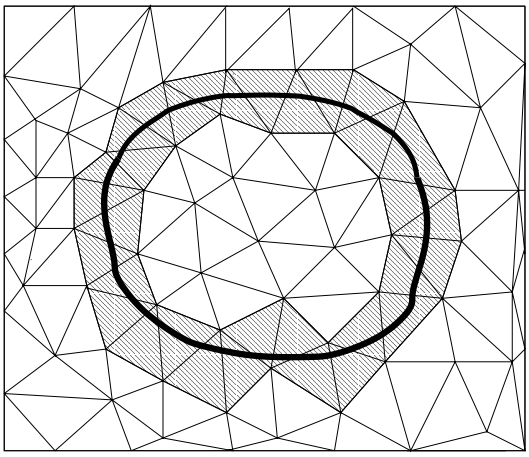


Figure 8.2: Computational domain split in two regions: interface region Ω_{int} (shaded) and far-field region Ω_{out} (white)

8.2.1 Interface local projection reinitialization algorithm

The idea underlying this method is based on a two step approach: we first find an *exact* discontinuous reinitialization of the level set function in Ω_{int} and then project it onto the piecewise linear finite element space. The notion of *exact* reinitialization will be clarified in the following section.

***Exact* discontinuous reinitialization**

First of all, we note that, in the case of $\mathbb{P}_1$ linear approximation, given $\widetilde{\phi}_h$, there exists a unique discontinuous piecewise linear reconstruction of the distance on each triangle $K_h \in \mathcal{T}_\Gamma$, given by:

$$d_h(\boldsymbol{x}) = \frac{\widetilde{\phi}_h(\boldsymbol{x})}{|\boldsymbol{\nabla}\widetilde{\phi}_h|}, \qquad \forall \boldsymbol{x} \in K_h, \tag{8.3}$$

where $\mathcal{T}_\Gamma$ is the restriction of the original triangulation $\mathcal{T}_h$ on Ω_{int} and $|\boldsymbol{\nabla}\widetilde{\phi}_h|$ is constant over each triangle.

The distance reconstruction defined in (8.3) is exact in the sense that the zero level sets of d_h and $\widetilde{\phi}_h$ coincide and $|\boldsymbol{\nabla} d_h| = 1$ on each $K_h \in \mathcal{T}_\Gamma$. Unfortunately, this reconstruction is discontinuous.

Projection onto the piecewise continuous linear finite element space

A linear continuous approximation $\phi_{h\text{int}}$ of the distance function in the interface region can be obtained by means of a L^2-projection in the following linear continuous finite element space:

$$W_h(\Omega_{\text{int}}) = \left\{ \phi_{h\text{int}} \in \mathcal{C}^0(\bar{\Omega_{\text{int}}}) \mid \phi_{h\text{int}}|_{K_h} \in \mathbb{P}_1(K_h),\ \forall K_h \in \mathcal{T}_\Gamma \right\}.$$

This is achieved through the solution of the following variational problem:

P8.1 *Find $\phi_{h\,int} \in W_h(\Omega_{int})$ such that*

$$\int_{\Omega_{int}} \phi_{h\,int}\, \psi_h \, d\boldsymbol{x} = \int_{\Omega_{int}} d_h\, \psi_h \, d\boldsymbol{x}, \quad \forall \psi_h \in W_h(\Omega_{int}).$$

Problem P(8.1) requires the solution of a mass algebraic problem whose size is proportional to the number of degrees of freedom involved in the interface region $\mathcal{T}_\Gamma$ and its computational cost is therefore negligible in the overall algorithm.

The distance function in the interface region defined in problem P(8.1) is the best continuous linear approximation of d_h in the L^2-sense. In the next section we will prove an a-priori error estimate for this approximation.

Far-field reinitialization

In the far-field region the accuracy of the reinitialization technique is less critical and any efficient method able to reconstruct a reasonable approximation of the distance function can be adopted. In this work, we consider, the Hamilton–Jacobi approach described in Section 8.1.2, using the solution of the interface local projection reinitialization as Dirichlet boundary condition. The Hamilton–Jacobi approach requires the solution of the following pseudo-time dependent problem:

$$\begin{cases} \dfrac{\partial \phi_{h\text{out}}}{\partial \tau} = \text{sign}(\widetilde{\phi}_h|_{\Omega_{\text{out}}})(1 - |\boldsymbol{\nabla}\phi_{h\text{out}}|), \\ \phi_{h\text{out}|\partial\Omega_{\text{out}}} = \phi_{h\text{int}}, \\ \phi_{h\text{out}|\tau=0} = \widetilde{\phi}_h|_{\Omega_{\text{out}}}, \end{cases} \tag{8.4}$$

where $\widetilde{\phi}_h|_{\Omega_{\text{out}}}$ is the restriction of $\widetilde{\phi}_h$ on Ω_{out}.

In this case the smoothing of the signum function is no more required since in each connected region $\widetilde{\phi}_h|_{\Omega_{\text{out}}}$ does not change sign. The variational form of problem (8.4) reads:

P8.2 *Find $\phi_{h\,out}(\boldsymbol{x},\tau) \in Y_h(\Omega_{out})$ such that, for any $\psi_h \in Y_{0,h}(\Omega_{out})$:*

$$\int_{\Omega_{out}} \frac{\partial \phi_{h\,out}}{\partial \tau}\psi_h \, dx = \int_{\Omega_{out}} sign(\widetilde{\phi}_h|_{\Omega_{out}})(1 - |\boldsymbol{\nabla}\phi_{h\,out}|)\, \psi_h \, dx,$$

where the finite element spaces are:

$$Y_h(\Omega_{\text{out}}) = \left\{\psi_h \in C^0(\Omega_{\text{out}})|\ \psi_h|_{K_h} \in \mathbb{P}_1, \psi_h|_{\partial\Omega_{\text{int}}} = \phi_{h\,\text{int}}|_{\partial\Omega_{\text{int}}}, \forall K_h \in \mathcal{T}_h \setminus \mathcal{T}_\Gamma\right\}$$

$$Y_{0,h}(\Omega_{\text{out}}) = \left\{\psi_h \in C^0(\Omega_{\text{out}})|\ \psi_h|_{K_h} \in \mathbb{P}_1, \psi_h|_{\partial\Omega_{\text{int}}} = 0, \forall K_h \in \mathcal{T}_h \setminus \mathcal{T}_\Gamma\right\}$$

Usually just a few iterations are sufficient to obtain a reasonable approximation of the distance function in the far-field region.

Finally, the reinitialized level set function is defined as:

$$\phi_h(\boldsymbol{x}) = \begin{cases} \phi_{h\,\text{int}}(\boldsymbol{x}) & \forall \boldsymbol{x} \in \Omega_{\text{int}} \\ \phi_{h\,\text{out}}(\boldsymbol{x}) & \forall \boldsymbol{x} \in \Omega_{\text{out}} \end{cases}$$

Remark. As already pointed out, the computational cost of the reinitialization in the interface region is negligible. The overall computational cost of the reinitialization mainly depend on the technique adopted in the far field region. Here, we have considered a Hamilton–Jacobi-type approach, but other techniques could also be designed to further improve the computational efficiency of the algorithm.

8.2.2 Abstract framework

At the continuous level, the reinitialization problem in the interface region Ω_{int} can be formulated by means of the following problem:

P8.3 *Given $\widetilde{\phi}$, find $\phi \in C^0$ such that*

$$\begin{aligned} |\boldsymbol{\nabla}\phi(\boldsymbol{x})| &= 1, && \forall \boldsymbol{x} \in \Omega_{int}, \\ sign(\phi) &= sign(\widetilde{\phi}), && \forall \boldsymbol{x} \in \Omega_{int}, \\ \phi|_\Gamma &= 0, \end{aligned}$$

where $\Gamma = \{\boldsymbol{x} \in \Omega_{\text{int}} \mid \widetilde{\phi}(\boldsymbol{x}) = 0\}$. Problem P8.3 states that ϕ is the signed distance function from the interface Γ and has the same zero level set as the non-reinitialized function $\widetilde{\phi}$.

We consider the following hypotheses for the solution ϕ of Problem P8.3: on each element $K_h \in \mathcal{T}_\Gamma$

(i) $\phi \in C^2(K_h)$;

(ii) $\boldsymbol{\nabla}\phi(\boldsymbol{x}_1) \cdot \boldsymbol{\nabla}\phi(\boldsymbol{x}_2) > 0, \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in K_h$;

(iii) There exist only two points $\boldsymbol{x}_A, \boldsymbol{x}_B \in \partial K_h$ such that $\phi(\boldsymbol{x}_A) = \phi(\boldsymbol{x}_B) = 0$;

Hypothesis *(iii)* yields that $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ are the only two intersections between Γ and the element boundary ∂K_h (see Fig. 8.3, left). In order to satisfy hypotheses *(i)-(iii)* the computational grid should be adequately refined in regions of high curvature of the interface (see Fig. 8.3, right). Moreover we assume that the triangulation $\mathcal{T}_\Gamma$ is quasi-uniform.

We first prove the following Poincaré–Friedrichs type inequality on the reference triangle $\widetilde{K}$.

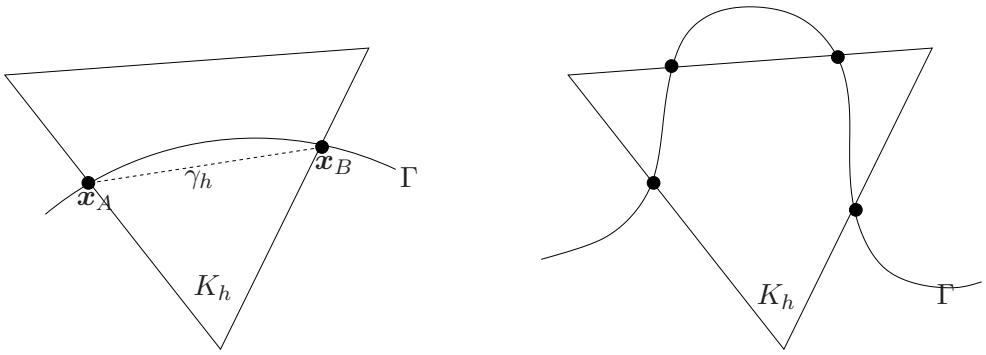


Figure 8.3: Admissible (left) and non admissible (right) interfacial element.

Lemma 8.2.1 *Given $\psi \in \mathcal{C}^1(\widetilde{K})$, if there exists $\boldsymbol{x}_0 \in \widetilde{K}$ such that $\psi(\boldsymbol{x}_0) = 0$, then*

$$\|\psi\|_{0,\widetilde{K}} \preceq \|\boldsymbol{\nabla}\psi\|_{0,\widetilde{K}}. \tag{8.5}$$

Proof Denoting with (x_0, y_0) the cartesian coordinates of $\boldsymbol{x}_0$, for each $\boldsymbol{x} = (x, y) \in \widetilde{K}$, we have

$$\psi(\boldsymbol{x}) = \int_{x_0}^{x} \psi_x d\xi + \int_{y_0}^{y} \psi_y d\eta. \tag{8.6}$$

By Cauchy–Schwarz inequality follows

$$\begin{aligned}
\|\psi\|^2_{0,\widetilde{K}} &= \int_{\widetilde{K}} \left(\int_{x_0}^{x} \psi_x \, d\xi + \int_{y_0}^{y} \psi_y \, d\eta \right)^2 d\boldsymbol{x} \\
&\preceq \int_{\widetilde{K}} \left(\int_{x_0}^{x} \psi_x \, d\xi \right)^2 + \left(\int_{y_0}^{y} \psi_y \, d\eta \right)^2 d\boldsymbol{x} \\
\text{(Cauchy–Schwarz)} \quad &\preceq \int_{\widetilde{K}} \left(\int_{x_0}^{x} 1 d\xi \int_{x_0}^{x} \psi_x^2 \, d\xi + \int_{y_0}^{y} 1 d\eta \int_{y_0}^{y} \psi_y^2 \, d\eta \right) d\boldsymbol{x} \\
&\preceq \int_{\widetilde{K}} \left(\int_{x_0}^{x} \psi_x^2 \, d\xi + \int_{y_0}^{y} \psi_y^2 \, d\eta \right) d\boldsymbol{x} \\
&= \int_0^1 \int_0^{1-y} \int_{x_0}^{x} \psi_x^2 \, d\xi \, dx \, dy + \int_0^1 \int_0^{1-x} \int_{y_0}^{y} \psi_y^2 \, d\eta \, dy \, dx.
\end{aligned} \tag{8.7}$$

We consider the first term in the right-hand-side of (8.7) and we have

$$\begin{aligned}
\int_0^1 \int_0^{1-y} \int_{x_0}^{x} \psi_x^2 \, d\xi \, dx \, dy &\leq \int_0^1 \int_0^{1-y} \int_0^{1-y} \psi_x^2 \, d\xi \, dx \, dy \\
&= \int_0^1 \int_0^{1-y} (1-y) \psi_x^2 \, d\xi \, dy \\
&\leq \int_0^1 \int_0^{1-y} \psi_x^2 \, d\xi \, dy = \|\psi_x\|^2_{0,\widetilde{K}}.
\end{aligned} \tag{8.8}$$

With an identical argument we can bound the second term in the right-hand-side of (8.7) to obtain

$$\int_0^1 \int_0^{1-x} \int_{y_0}^{y} \psi_y^2 \, d\eta \, dy \, dx \leq \|\psi_y\|_{0,\widetilde{K}}^2. \tag{8.9}$$

From (8.7), (8.8) and (8.9), we can prove inequality (8.5), as follows

$$\|\psi\|_{0,\widetilde{K}} \preceq \left(\|\psi_x\|_{0,\widetilde{K}}^2 + \|\psi_y\|_{0,\widetilde{K}}^2 \right)^{1/2} = \|\boldsymbol{\nabla}\psi\|_{0,\widetilde{K}}. \tag{8.10}$$

■

Corollary 8.2.1 *On the generic triangle $K_h \in \mathcal{T}_\Gamma$, under the same hypothesis of Lemma 8.2.1, we have*

$$\|\psi\|_{0,K} \preceq h \|\boldsymbol{\nabla}\psi\|_{0,K} \tag{8.11}$$

Proof The proof is a direct consequence of a standard scaling argument as the one used in (7.18).

■

We introduce now the following discontinuous piecewise linear space

$$W_h^D(\Omega_{\text{int}}) = \left\{ \phi_{h\,\text{int}} \in L^2(\Omega_{\text{int}}) |\, \phi_{h\,\text{int}}|_{K_h} \in \mathbb{P}_1(K_h),\ \forall K_h \in \mathcal{T}_\Gamma \right\}.$$

and we consider a linear discontinuous interpolation operator $\mathcal{I}_h^D$.

Definition 1 *Given $\phi \in \mathcal{C}^2(K_h), \forall K_h \in \mathcal{T}_\Gamma$, the interpolation operator $\mathcal{I}_h^D$ is uniquely defined by requiring on each element $K_h \in \mathcal{T}_\Gamma$ that the following properties hold:*

$$\mathcal{I}_h^D \phi(\boldsymbol{x}_A) = 0, \tag{8.12}$$

$$\mathcal{I}_h^D \phi(\boldsymbol{x}_B) = 0, \tag{8.13}$$

$$\left| \boldsymbol{\nabla} \mathcal{I}_h^D \phi \right| = 1, \tag{8.14}$$

$$\boldsymbol{\nabla} \mathcal{I}_h^D \phi \cdot \boldsymbol{\nabla}\phi > 0. \tag{8.15}$$

In the following Lemma we show the approximation properties of $\mathcal{I}_h^D$.

Lemma 8.2.2 *Given $\phi \in \mathcal{C}^2(K_h), \forall K_h \in \mathcal{T}_\Gamma$, under hypotheses (i)-(iii), the following estimates hold:*

$$\|\phi - \mathcal{I}_h^D \phi\|_{0,\Omega_{int}} \preceq h^2 \, \|\phi\|_{2,\Omega_{int}}, \tag{8.16}$$

$$\|\boldsymbol{\nabla}\phi - \boldsymbol{\nabla}\mathcal{I}_h^D \phi\|_{0,\Omega_{int}} \preceq h \, \|\phi\|_{2,\Omega_{int}}. \tag{8.17}$$

Proof We claim that, if $\phi \in \mathcal{C}^2(K_h)$, then there exists $\boldsymbol{\xi} \in K_h$ such that

$$\boldsymbol{\nabla} \mathcal{I}_h^D \phi(\boldsymbol{\xi}) = \boldsymbol{\nabla}\phi(\boldsymbol{\xi}). \tag{8.18}$$

To prove (8.18), we first consider the relation

$$\phi(\boldsymbol{x}_A) - \phi(\boldsymbol{x}_B) = \int_{\gamma_h} \boldsymbol{\nabla}\phi \cdot \boldsymbol{\tau}_{\gamma_h} d\sigma = 0, \tag{8.19}$$

where γ_h is the segment connecting $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ and $\boldsymbol{\tau}_{\gamma_h}$ is the tangential unit vector along γ_h. By the mean value theorem and using the regularity of ϕ, we have that $\exists \boldsymbol{\xi} \in \gamma_h$ such that:

$$\boldsymbol{\nabla}\phi(\boldsymbol{\xi}) \cdot \boldsymbol{\tau}_{\gamma_h}(\boldsymbol{\xi}) = 0. \tag{8.20}$$

Moreover, since $\mathcal{I}_h^D\phi$ is linear on each K_h, we have $\forall \boldsymbol{x} \in K_h$:

$$\boldsymbol{\nabla}\mathcal{I}_h^D\phi(\boldsymbol{x}) \cdot \boldsymbol{\tau}_{\gamma_h}(\boldsymbol{x}) = 0. \tag{8.21}$$

Since $|\boldsymbol{\nabla}\phi| = |\boldsymbol{\nabla}\mathcal{I}_h^D\phi| = 1$, (8.20)-(8.21) imply the following identities:

$$\boldsymbol{\nabla}\phi(\boldsymbol{\xi}) \cdot \boldsymbol{n}_{\gamma_h}(\boldsymbol{\xi}) \;=\; 1, \tag{8.22}$$

$$\boldsymbol{\nabla}\mathcal{I}_h^D\phi(\boldsymbol{x}) \cdot \boldsymbol{n}_{\gamma_h}(\boldsymbol{x}) \;=\; 1, \qquad \forall \boldsymbol{x} \in K_h \tag{8.23}$$

where $\boldsymbol{n}_{\gamma_h}$ is the unit vector normal to γ_h. Identity (8.18) is a direct consequence of relations (8.20)-(8.23) and property (8.15).

Now, applying Corollary (8.2.1) componentwise to the function $\boldsymbol{\nabla}\phi - \boldsymbol{\nabla}\mathcal{I}_h^D\phi$ and using the fact that $\boldsymbol{\nabla}\mathcal{I}_h^D\phi$ is constant on K_h, we can derive an H^1 interpolation error for $\mathcal{I}_h^D$:

$$\|\boldsymbol{\nabla}\phi - \boldsymbol{\nabla}\mathcal{I}_h^D\phi\|_{0,K_h} \preceq h\,\|\boldsymbol{D}^2\phi\|_{0,K_h} \leq h\,\|\phi\|_{2,K_h}. \tag{8.24}$$

Analogously, applying the result of Corollary (8.2.1) to the function $\phi - \mathcal{I}_h^D\phi$ and using (8.24), we obtain an L_2 interpolation error for $\mathcal{I}_h^D$, as follows

$$\|\phi - \mathcal{I}_h^D\phi\|_{0,K_h}^2 \preceq h\,\|\boldsymbol{\nabla}\phi - \boldsymbol{\nabla}\mathcal{I}_h^D\phi\|_{0,K_h} \preceq h^2\,\|\phi\|_{2,K_h}. \tag{8.25}$$

The proof is completed showing that inequalities (8.24) and (8.25) imply, respectively:

$$\begin{aligned}
\|\boldsymbol{\nabla}\phi - \boldsymbol{\nabla}\mathcal{I}_h^D\phi\|_{0,\Omega_{\text{int}}} &\leq \left(\sum_{K_h} \|\boldsymbol{\nabla}\phi - \boldsymbol{\nabla}\mathcal{I}_h^D\phi\|_{0,K_h}^2\right)^{1/2} \\
&\preceq \left(\sum_{K_h} h^2 \|\boldsymbol{D}^2(\phi)\|_{0,K_h}^2\right)^{1/2} \\
&\preceq h\,\|\phi\|_{2,\Omega_{\text{int}}},
\end{aligned} \tag{8.26}$$

and

$$\begin{aligned}
\|\phi - \mathcal{I}_h^D\phi\|_{0,\Omega_{\text{int}}} &\leq \left(\sum_{K_h} \|\phi - \mathcal{I}_h^D\phi\|_{0,K_h}^2\right)^{1/2} \\
&\leq \left(\sum_{K_h} C\,h^4 \|\boldsymbol{D}^2(\phi)\|_{0,K_h}^2\right)^{1/2} \\
&\leq Ch^2\,\|\phi\|_{2,\Omega_{\text{int}}}.
\end{aligned} \tag{8.27}$$

■

The discontinuous interpolant operator $\mathcal{I}_h^D$ therefore satisfies the optimal interpolation error estimates (8.16) and (8.17). Indeed, $\mathcal{I}_h^D\phi$ represents the best local linear approximation of the solution ϕ to the reinitialization problem P8.3.

We now consider a continuous interpolant operator $\mathcal{I}_h^C$ defined by

$$\mathcal{I}_h^C\phi = \pi_h\mathcal{I}_h^D\phi, \tag{8.28}$$

where π_h is the standard L^2-projection operator on the subspace $W_h(\Omega_{\text{int}})$. Therefore, we have

$$\left(\mathcal{I}_h^C\phi - \mathcal{I}_h^D\phi, v_h\right) = 0, \qquad \forall v_h \in W_h(\Omega_{\text{int}}). \tag{8.29}$$

The convergence properties of $\mathcal{I}_h^C$ are given in the following Lemma.

Lemma 8.2.3 *Under hypotheses (i)-(iii), the following error estimates hold*

$$\|\phi - \mathcal{I}_h^C\phi\|_{0,\Omega_{int}} \preceq h^2\|\phi\|_{2,\Omega_{int}}, \tag{8.30}$$

$$\|\boldsymbol{\nabla}\phi - \boldsymbol{\nabla}\mathcal{I}_h^C\phi\|_{0,\Omega_{int}} \preceq h\|\phi\|_{2,\Omega_{int}}. \tag{8.31}$$

Proof Using the triangular inequality, we have:

$$\|\phi - \mathcal{I}_h^C\phi\|_{0,\Omega_{\text{int}}} \leq \|\phi - \mathcal{I}_h^D\phi\|_{0,\Omega_{\text{int}}} + \|\mathcal{I}_h^C\phi - \mathcal{I}_h^D\phi\|_{0,\Omega_{\text{int}}} \tag{8.32}$$

The first term in the right hand side of (8.32) is bounded by (8.16).

To bound the second term, we first introduce the quasi-interpolant $\mathcal{I}_h^{\text{Os}}$ due to Oswald [Osw91], which is defined as follows: given $\psi_h \in W_h^D(\Omega_{\text{int}})$

$$\mathcal{I}_h^{\text{Os}}\psi_h(\boldsymbol{x}_i) = \frac{1}{n_i}\sum_{K_h\in\Omega_i}\psi_h|_{K_h}(\boldsymbol{x}_i),$$

where Ω_i is the patch of elements which share node $\boldsymbol{x}_i$ and n_i in the number of these elements. For the Oswald interpolation the following property holds (see, *e.g.*, [HW96, ABC03, KP03]):

$$\|\mathcal{I}_h^{\text{Os}}\mathcal{I}_h^D\phi - \mathcal{I}_h^D\phi\|_{0,\Omega_{\text{int}}}^2 \preceq \sum_{K_h}\int_{\partial K_h\setminus\partial\Omega_{\text{int}}} h[\mathcal{I}_h^D\phi]^2\,d\sigma \tag{8.33}$$

Using property (8.29), we have:

$$\begin{aligned}
\|\mathcal{I}_h^C\phi - \mathcal{I}_h^D\phi\|_{0,\Omega_{\text{int}}}^2 &= \left(\mathcal{I}_h^C\phi - \mathcal{I}_h^D\phi, \mathcal{I}_h^C\phi - \mathcal{I}_h^D\phi\right)\\
&= \left(\mathcal{I}_h^C\phi - \mathcal{I}_h^D\phi, \mathcal{I}_h^{\text{Os}}\mathcal{I}_h^D\phi - \mathcal{I}_h^D\phi\right)\\
&\leq \|\mathcal{I}_h^C\phi - \mathcal{I}_h^D\phi\|_{0,\Omega_{\text{int}}}\|\mathcal{I}_h^{\text{Os}}\mathcal{I}_h^D\phi - \mathcal{I}_h^D\phi\|_{0,\Omega_{\text{int}}}.
\end{aligned} \tag{8.34}$$

Moreover,

$$\begin{aligned}
\|\mathcal{I}_h^C\phi - \mathcal{I}_h^D\phi\|_{0,\Omega_{\text{int}}} &\leq \|\mathcal{I}_h^{\text{Os}}\mathcal{I}_h^D\phi - \mathcal{I}_h^D\phi\|_{0,\Omega_{\text{int}}}\\
&\preceq \left(\sum_{K_h}\int_{\partial K_h\setminus\partial\Omega_{\text{int}}} h[\mathcal{I}_h^D\phi]^2\,d\sigma\right)^{1/2}\\
&= \left(\sum_{K_h}\int_{\partial K_h\setminus\partial\Omega_{\text{int}}} h[\mathcal{I}_h^D\phi - \phi]^2\,d\sigma\right)^{1/2},
\end{aligned} \tag{8.35}$$

where we have used (8.33) and the fact that ϕ is continuous across $\partial K_h \setminus \partial\Omega_{\text{int}}$, namely the internal edges of $\mathcal{T}_\Gamma$.

By means of the trace inequality (7.19) and using the estimates (8.16)-(8.17) obtained in Lemma 8.2.2, we can bound the right hand side of (8.35), as follows:

$$\sum_{K_h} \int_{\partial K_h \setminus \partial\Omega_{\text{int}}} h[\mathcal{I}_h^D\phi - \phi]^2 \, d\sigma \preceq \left(\sum_{K_h} \int_{K_h} \|\mathcal{I}_h^D\phi - \phi\|_{0,K_h}^2 + \sum_{K_h} h^2 \int_{K_h} \|\mathcal{I}_h^D\phi - \phi\|_{1,K_h}^2 \right)$$
$$\preceq h^4 \|\phi\|_{2,\Omega_{\text{int}}}^2. \tag{8.36}$$

Combining (8.35) and (8.36), we have

$$\|\mathcal{I}_h^C\phi - \mathcal{I}_h^D\phi\|_{0,\Omega_{\text{int}}} \preceq h^2 \|\phi\|_{2,\Omega_{\text{int}}}, \tag{8.37}$$

and the optimal L^2 estimate (8.30) results from (8.32) together with (8.16) and (8.37).

The optimal H^1 estimate (8.31) is finally obtained, using a similar argument, as follows:

$$\begin{aligned} \|\boldsymbol{\nabla}(\phi - \mathcal{I}_h^C\phi)\|_{0,\Omega_{\text{int}}} &\preceq \|\boldsymbol{\nabla}(\phi - \mathcal{I}_h^D\phi)\|_{0,\Omega_{\text{int}}} + \|\boldsymbol{\nabla}(\mathcal{I}_h^C\phi - \mathcal{I}_h^D\phi)\|_{0,\Omega_{\text{int}}} \\ &\preceq \left(h\|\phi\|_{2,\Omega_{\text{int}}} + \frac{1}{h}\|(\mathcal{I}_h^C\phi - \mathcal{I}_h^D\phi)\|_{0,\Omega_{\text{int}}} \right) \\ &\preceq h\|\phi\|_{2,\Omega_{\text{int}}}. \end{aligned} \tag{8.38}$$

■

Moreover, an error estimate in the L^2-norm on the interface Γ can be obtained as done in following Corollary.

Corollary 8.2.2 *The following convergence estimate holds:*

$$\|\phi - \mathcal{I}_h^C\phi\|_{0,\Gamma} \preceq h^{3/2} \|\phi\|_{2,\Omega_{int}}. \tag{8.39}$$

Proof We consider the following trace inequality:

$$\|\phi\|_{0,\Gamma} \preceq h^{-1/2}\|\phi\|_{0,\Omega_{\text{int}}} + h^{1/2}\|\boldsymbol{\nabla}\phi\|_{0,\Omega_{\text{int}}}. \tag{8.40}$$

Using (8.40) and estimates (8.30) and (8.30), we have

$$\begin{aligned} \|\phi - \mathcal{I}_h^C\phi\|_{0,\Gamma} &\preceq h^{-1/2}\|\phi - \mathcal{I}_h^C\phi\|_{0,\Omega_{\text{int}}} + h^{1/2}\|\boldsymbol{\nabla}(\phi - \mathcal{I}_h^C\phi)\|_{0,\Omega_{\text{int}}} \\ &\preceq h^{-1/2}(h^2\|\phi\|_{2,\Omega_{\text{int}}}) + h^{1/2}(h\|\phi\|_{2,\Omega_{\text{int}}}) \\ &\preceq h^{3/2}\|\phi\|_{2,\Omega_{\text{int}}}. \end{aligned}$$

■

8.2.3 Convergence analysis of the interface local projection reinitialization

We now consider the solution ϕ_h to problem P8.1, where we have dropped the subscript "int" for ease of exposition. Given $\widetilde{\phi}_h$, the function ϕ_h is the discrete level set function in Ω_{int} reinitialized by the interface local projection procedure.

We assume that $\widetilde{\phi}_h$ satisfies the following optimal error estimates

$$\|\widetilde{\phi}_h - \widetilde{\phi}\|_{0,\Omega_{\text{int}}} \preceq h^2 \|\widetilde{\phi}\|_{2,\Omega_{\text{int}}}, \tag{8.41}$$

$$\|\boldsymbol{\nabla}(\widetilde{\phi}_h - \widetilde{\phi})\|_{0,\Omega_{\text{int}}} \preceq h \|\widetilde{\phi}\|_{2,\Omega_{\text{int}}}. \tag{8.42}$$

Using the interpolation operator $\mathcal{I}_h^C$ introduced in the previous section, we have:

$$\phi_h = \mathcal{I}_h^C \widetilde{\phi}_h. \tag{8.43}$$

We have defined the interface Γ as the zero level set of $\widetilde{\phi}$. By definition, Γ is also the zero level set of ϕ, solution of the continuous reinitialization problem P8.3. Analogously, we introduce two discrete approximations of the interface defined, respectively, as the zero level set of $\widetilde{\phi}_h$ and ϕ_h, namely

$$\widetilde{\Gamma}_h = \{\boldsymbol{x} \in \Omega_{\text{int}} \mid \widetilde{\phi}_h(\boldsymbol{x}) = 0\},$$

$$\Gamma_h = \{\boldsymbol{x} \in \Omega_{\text{int}} \mid \phi_h(\boldsymbol{x}) = 0\}.$$

We introduce the notion of interface convergence as follows:

Definition 2 *Consider a family of functions $\{\phi_\epsilon\}$ and a function ϕ and assume that there exists a constant $c > 0$ such that $|\boldsymbol{\nabla}\phi_\epsilon| > c, \forall \epsilon$, and $|\boldsymbol{\nabla}\phi| > c$. Denoting by Γ_ϵ and Γ the zero level sets of ϕ_ϵ and ϕ, respectively, we say that Γ_ϵ converge to Γ, if $\|\phi_\epsilon\|_{0,\Gamma} \longrightarrow 0$ as $\epsilon \longrightarrow 0$. The convergence is denoted by $\Gamma_\epsilon \longrightarrow \Gamma$.*

A convergence result on the approximation of the interface is given in the following Lemma.

Lemma 8.2.4 *If there exists a constant $c > 0$ such that $|\boldsymbol{\nabla}\widetilde{\phi}_h| > c$, then the interface approximation Γ_h converges to Γ.*

Proof Using the trace inequality (8.40) and the error estimates (8.41) and (8.42), we have

$$\begin{aligned}
\|\widetilde{\phi} - \widetilde{\phi}_h\|_{0,\widetilde{\Gamma}_h} &\preceq h^{-1/2}\|\widetilde{\phi} - \widetilde{\phi}_h\|_{0,\Omega_{\text{int}}} + h^{1/2}\|\boldsymbol{\nabla}(\widetilde{\phi} - \widetilde{\phi}_h)\|_{0,\Omega_{\text{int}}} \\
&\preceq h^{-1/2}(h^2\|\widetilde{\phi}\|_{2,\Omega_{\text{int}}}) + h^{1/2}(h\|\widetilde{\phi}\|_{2,\Omega_{\text{int}}}) \\
&\preceq h^{3/2}\|\widetilde{\phi}\|_{2,\Omega_{\text{int}}},
\end{aligned} \tag{8.44}$$

which implies that

$$\widetilde{\Gamma}_h \longrightarrow \Gamma, \qquad \textit{as } h \longrightarrow 0, \tag{8.45}$$

since the zero level sets of $\widetilde{\phi}$ and ϕ coincide.

We then have to show that

$$\Gamma_h \longrightarrow \widetilde{\Gamma}_h, \qquad \textit{as } h \longrightarrow 0, \tag{8.46}$$

or, equivalently, that

$$\|\phi_h\|_{0,\widetilde{\Gamma}_h} \longrightarrow 0, \qquad \textit{as} \;\; h \longrightarrow 0. \tag{8.47}$$

To prove (8.47), we first note that:

$$\|\phi_h - \widetilde{\phi}_h\|_{0,\widetilde{\Gamma}_h} = \|\phi_h - \mathcal{I}_h^D \widetilde{\phi}_h\|_{0,\widetilde{\Gamma}_h}, \tag{8.48}$$

since, by the definition of the interpolant operator $\mathcal{I}_h^D$ (see Def. 1), we have

$$\mathcal{I}_h^D \widetilde{\phi}_h|_{\widetilde{\Gamma}_h} = \widetilde{\phi}_h|_{\widetilde{\Gamma}_h}.$$

Using the trace inequality (8.40) and the inverse inequality, we have

$$\begin{aligned}
\|\phi_h\|^2_{0,\widetilde{\Gamma}_h} &= \|\phi_h - \widetilde{\phi}_h\|^2_{0,\widetilde{\Gamma}_h} = \|\phi_h - \mathcal{I}_h^D \widetilde{\phi}_h\|^2_{0,\widetilde{\Gamma}_h} \\
&\preceq h^{-1}\|\phi_h - \mathcal{I}_h^D \widetilde{\phi}_h\|^2_{0,\Omega_{\text{int}}} + h\|\phi_h - \mathcal{I}_h^D \widetilde{\phi}_h\|^2_{1,\Omega_{\text{int}}} \\
&\preceq h^{-1}\|\phi_h - \mathcal{I}_h^D \widetilde{\phi}_h\|^2_{0,\Omega_{\text{int}}}.
\end{aligned}$$

Moreover, using the Galerkin orthogonality and estimates (8.16)-(8.17), we obtain

$$\begin{aligned}
\|\phi_h\|^2_{0,\widetilde{\Gamma}_h} &= \|\phi_h - \widetilde{\phi}_h\|^2_{0,\widetilde{\Gamma}_h} \preceq h^{-1}\|\phi_h - \mathcal{I}_h^D \widetilde{\phi}_h\|^2_{0,\Omega_{\text{int}}} \\
&= h^{-1}(\phi_h - \mathcal{I}_h^D \widetilde{\phi}_h, \phi_h - \mathcal{I}_h^D \widetilde{\phi}_h)_{0,\Omega_{\text{int}}} \\
&= h^{-1}(\phi_h - \mathcal{I}_h^D \widetilde{\phi}_h, \mathcal{I}_h^{\text{Os}} \mathcal{I}_h^D \widetilde{\phi}_h - \mathcal{I}_h^D \widetilde{\phi}_h)_{0,\Omega_{\text{int}}} \\
&\leq h^{-1}\|\phi_h - \mathcal{I}_h^D \widetilde{\phi}_h\|_{0,\Omega_{\text{int}}} \|\mathcal{I}_h^{\text{Os}} \mathcal{I}_h^D \widetilde{\phi}_h - \mathcal{I}_h^D \widetilde{\phi}_h\|_{0,\Omega_{\text{int}}}.
\end{aligned} \tag{8.49}$$

Inequality (8.49), combined with Oswald interpolation property (8.33), yields

$$\begin{aligned}
\|\phi_h - \widetilde{\phi}_h\|^2_{0,\widetilde{\Gamma}_h} &\preceq h^{-1}\|\mathcal{I}_h^{\text{Os}} \mathcal{I}_h^D \widetilde{\phi}_h - \mathcal{I}_h^D \widetilde{\phi}_h\|^2_{0,\Omega_{\text{int}}} \\
&\preceq h^{-1} \sum_{K_h} \int_{\partial K_h \setminus \partial\Omega_{\text{int}}} h[\mathcal{I}_h^D \widetilde{\phi}_h]^2 \, d\sigma \\
&\preceq \sum_{K_h} \int_{\partial K_h \setminus \partial\Omega_{\text{int}}} [\mathcal{I}_h^D \widetilde{\phi}_h]^2 \, d\sigma \\
&\preceq \sum_{K_h} \int_{\partial K_h \setminus \partial\Omega_{\text{int}}} h^2 [\boldsymbol{\nabla} \mathcal{I}_h^D \widetilde{\phi}_h]^2 \, d\sigma.
\end{aligned} \tag{8.50}$$

where we have used the fact that $\exists \boldsymbol{x}_0 \in \partial K_h \setminus \partial\Omega_{\text{int}}$ such that $\mathcal{I}_h^D \widetilde{\phi}_h(\boldsymbol{x}_0) = 0$ and

$$\begin{aligned}
[\mathcal{I}_h^D \widetilde{\phi}_h(\boldsymbol{x})] &= [\mathcal{I}_h^D \widetilde{\phi}_h(\boldsymbol{x}_0)] + \int_{\boldsymbol{x}_0}^{\boldsymbol{x}} (\boldsymbol{\xi} - \boldsymbol{x}_0)[\boldsymbol{\nabla} \mathcal{I}_h^D \widetilde{\phi}_h] d\xi \\
&= \int_{\boldsymbol{x}_0}^{\boldsymbol{x}} (\boldsymbol{\xi} - \boldsymbol{x}_0)[\boldsymbol{\nabla} \mathcal{I}_h^D \widetilde{\phi}_h] d\xi, \qquad \forall \boldsymbol{x} \in \partial K_h \setminus \partial\Omega_{\text{int}}.
\end{aligned}$$

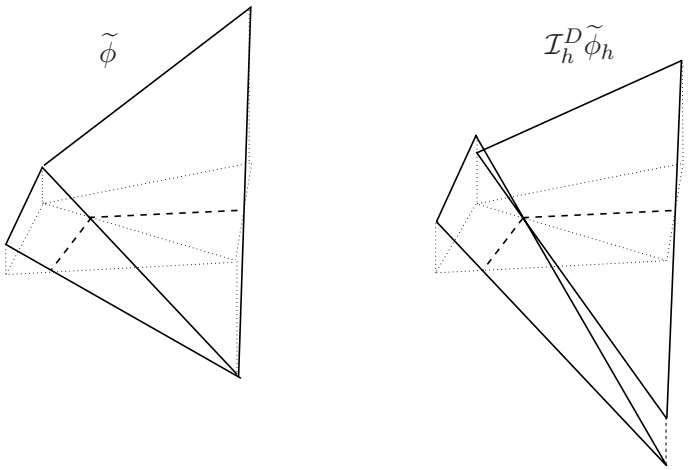


Figure 8.4: The continuous function $\widetilde{\phi}_h$ and its discontinuous interpolate $\mathcal{I}_h^D\widetilde{\phi}_h$, with $|\boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h| = 1$ on each $K_h \in \mathcal{T}_h$ (the interface segments are the dashed lines).

Now, we claim that the jump of the gradient of the discontinuous function $\mathcal{I}_h^D\widetilde{\phi}_h$ (see Fig 8.4, right) can be bounded by the jump of the gradient of $\widetilde{\phi}_h$ (see Fig 8.4, left), namely

$$[\boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h]^2 \preceq [\boldsymbol{\nabla}\widetilde{\phi}_h]^2. \tag{8.51}$$

To prove (8.51), we first evaluate $[\boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h]^2$, on the edge shared by elements K_1 and K_2. We can show that it only depends on the angle α between the two interface segments in K_1 and K_2. Indeed, if we consider a local reference system aligned with the interface segment in K_1 (see Fig. 8.5), we have

$$\begin{aligned}
[\boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h]^2 &= \Big((\boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h|_{T_1} - \boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h|_{T_2})\cdot\boldsymbol{n}_1\Big)^2 + \Big((\boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h|_{T_1} - \boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h|_{T_2})\cdot\boldsymbol{t}_1\Big)^2 \\
&= \Big(1 - \boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h|_{T_2}\cdot\boldsymbol{n}_1\Big)^2 + \Big(\boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h|_{T_2}\cdot\boldsymbol{t}_1\Big)^2 \\
&= 1 - 2\cos(\alpha) + (\boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h|_{T_2}\cdot\boldsymbol{n}_1)^2 + (\boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h|_{T_2}\cdot\boldsymbol{t}_1)^2 \\
&= 2 - 2\cos(\alpha),
\end{aligned} \tag{8.52}$$

where we have used the fact that $\boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h|_{T_1}\cdot\boldsymbol{t}_1 = 0$ and $\boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h|_{T_1}\cdot\boldsymbol{n}_1 = |\boldsymbol{\nabla}\mathcal{I}_h^D\widetilde{\phi}_h|_{T_1}| = 1$.

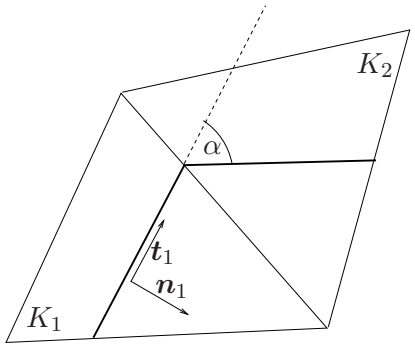


Figure 8.5: Local reference system aligned with the interface on K_1.

Analogously, we now consider $[\nabla\widetilde{\phi}_h]$ across the edge shared by elements K_1 and K_2. Using the hypothesis on the boundedness of $\nabla\phi_h$, we can prove (8.51):

$$
\begin{aligned}
[\nabla\widetilde{\phi}_h]^2 &= \left((\nabla\widetilde{\phi}_h|_{T_1} - \nabla\widetilde{\phi}_h|_{T_2})\cdot\boldsymbol{n}_1\right)^2 + \left((\nabla\widetilde{\phi}_h|_{T_1} - \nabla\widetilde{\phi}_h|_{T_2})\cdot\boldsymbol{t}_1\right)^2 \\
&= \left(\nabla\widetilde{\phi}_h|_{T_1}\cdot\boldsymbol{n}_1 - \nabla\widetilde{\phi}_h|_{T_2}\cdot\boldsymbol{n}_1\right)^2 + \left(\nabla\widetilde{\phi}_h|_{T_2}\cdot\boldsymbol{t}_1\right)^2 \\
&= \left|\nabla\widetilde{\phi}_h|_{T_1}\right|^2 + (\nabla\widetilde{\phi}_h|_{T_2}\cdot\boldsymbol{n}_1)^2 - 2\left|\nabla\widetilde{\phi}_h|_{T_1}\right|\left|\nabla\widetilde{\phi}_h|_{T_2}\right|\cos(\alpha) + (\nabla\widetilde{\phi}_h|_{T_2}\cdot\boldsymbol{t}_1)^2 \\
&= \left|\nabla\widetilde{\phi}_h|_{T_1}\right|^2 + \left|\nabla\widetilde{\phi}_h|_{T_2}\right|^2 - 2\left|\nabla\widetilde{\phi}_h|_{T_1}\right|\left|\nabla\widetilde{\phi}_h|_{T_2}\right|\cos(\alpha) \qquad (8.53) \\
&= \left(\left|\nabla\widetilde{\phi}_h|_{T_1}\right| - \left|\nabla\widetilde{\phi}_h|_{T_2}\right|\right)^2 + \left|\nabla\widetilde{\phi}_h|_{T_1}\right|\left|\nabla\widetilde{\phi}_h|_{T_2}\right|(2-2\cos(\alpha)) \\
&\geq \left|\nabla\widetilde{\phi}_h|_{T_1}\right|\left|\nabla\widetilde{\phi}_h|_{T_2}\right|(2-2\cos(\alpha)) \\
&\succeq [\nabla\mathcal{I}_h^D\widetilde{\phi}_h]^2,
\end{aligned}
$$

where we have used the fact that $\nabla\widetilde{\phi}_h|_{T_1}\cdot\boldsymbol{t}_1 = 0$.

The convergence property (8.47) is obtained combining (8.50) and (8.51) and proceeding as in (8.36), as follows

$$
\begin{aligned}
\|\phi_h\|_{0,\widetilde{\Gamma}_h} &= \|\phi_h - \widetilde{\phi}_h\|_{0,\widetilde{\Gamma}_h} \\
&\preceq \left(\sum_{K_h}\int_{\partial K_h\setminus\partial\Omega_{\text{int}}} h^2[\nabla\widetilde{\phi}_h]^2\,d\sigma\right)^{1/2} \\
&= \left(\sum_{K_h}\int_{\partial K_h\setminus\partial\Omega_{\text{int}}} h^2[\nabla(\widetilde{\phi}_h - \widetilde{\phi})]^2\,d\sigma\right)^{1/2} \qquad (8.54) \\
&\preceq \left(\sum_{K_h}\left(h\|\nabla(\widetilde{\phi}_h - \widetilde{\phi})\|^2_{0,K_h} + h^3\|\widetilde{\phi}\|_{2,K_h}\right)\right)^{1/2} \\
&\preceq h^{3/2}\|\widetilde{\phi}\|_{2,\Omega_{\text{int}}},
\end{aligned}
$$

where we have used estimate (8.42) and the fact that $\nabla\widetilde{\phi}$ is continuous across the internal edges of $\mathcal{T}_\Gamma$.

■

Remark. The hypothesis on the boundedness of $\nabla\widetilde{\phi}_h$ at the interface is not very restrictive in practice. In fact, the reinitialization procedure in level set computations is operated every m time iterations, with m depending on the specific problem (usually $m \approx 10$). During these m iterations, we can assume that the gradient of $\widetilde{\phi}$ does not have time to degenerate to zero. Indeed, this is exactly the role of the reinitialization, *i.e.* avoiding the flattening (as well as the steepening) of the level set function close to the interface.

Remark. The convergence estimates (8.44) and (8.54) obtained in Lemma 8.2.4 are optimal. In the level set method, the reinitialization procedure introduced above is applied to the function$\widetilde{\phi}_h$

which is the discrete solution of the advection problem (6.25). If the latter is solved by means of a stabilized Galerkin approach, as the one introduced in Chapter 7, only sub-optimal error estimates on $\widetilde{\phi}_h$ hold (see (7.12)-(7.13), with $k = 1$). In this case, weaker convergence results for the reinitialization procedure can be obtained.

8.3 Numerical test cases

In this section, we present some numerical tests where the interface local projection reinitialization has been used, comparing its behaviour with other reinitialization techniques. In particular, we will analyse the impact of the reinitialization on the mass conservation and its role in the solution of two-fluid flows.

8.3.1 Mass conservation test

As already pointed out, the reinitialization procedure can influence the global mass conservation properties of the numerical scheme, with the obvious negative consequences when two-fluid flow problems are considered. In order to assess the accuracy of the different reinitialization techniques described in Sections 8.1 and 8.2, we have performed specific numerical tests measuring the effect of the reinitialization on the mass conservation. We consider three different initial interface shapes: a circle, a square and a non-convex shape obtained as the boundary of the intersection of two circles (see Fig. 8.6).

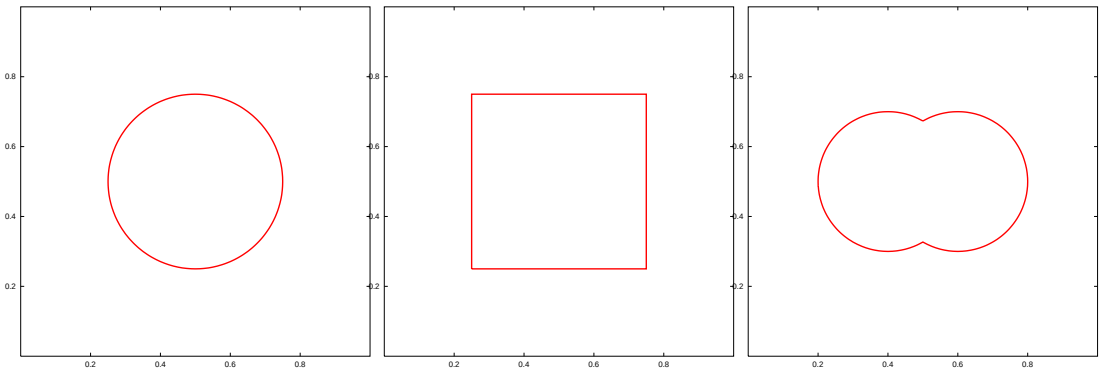


Figure 8.6: Interface shapes considered in the mass conservation test case

For each case, the level set function is initialized as the double of the signed distance function from the interface. The direct, Hamilton–Jacobi and interface local projection procedures have been used for the reinitialization. The initial and reinitialized level set functions associated to the three interface shapes are shown in Figs. 8.7, 8.8 and 8.9.

Tables 8.1-8.3 present the results obtained using the three reinitializations on the different interface shapes. The area inside the interface is measured and compared with the analytical value. The relative errors obtained using different uniform unstructured grids (with $h = 1/20, 1/40, 1/80, 1/160$) and the convergence rates are presented. We can note that the convergence rates for direct and interface local projection reinitializations are comparable (≈ 2). On the other hand, the Hamilton–Jacobi approach shows a lower convergence rate (≈ 1). It should also be noted that relative error

obtained using the interface local projection reinitialization is consistently (from 3 to 15 times) lower than the one obtained with the direct method.

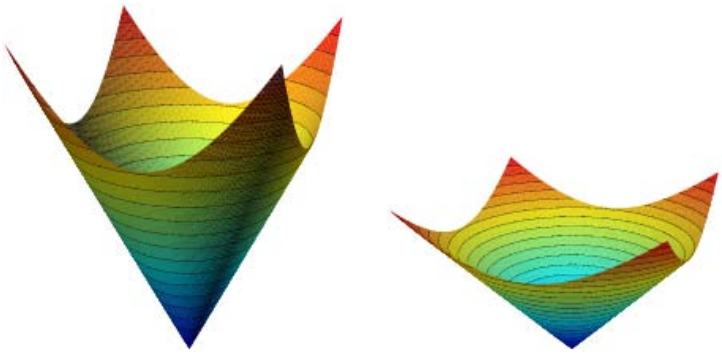


Figure 8.7: Circular interface: before reinitialization (left), after reintialization (right).

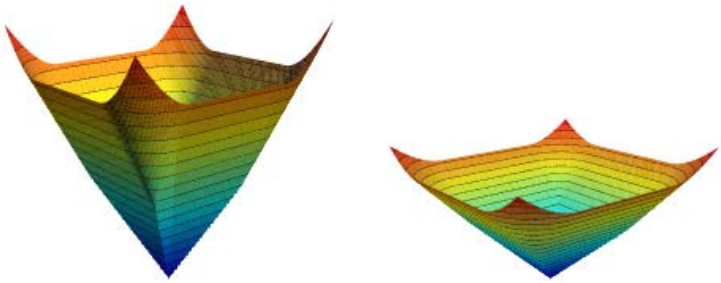


Figure 8.8: Square interface: before reinitialization (left), after reintialization (right).

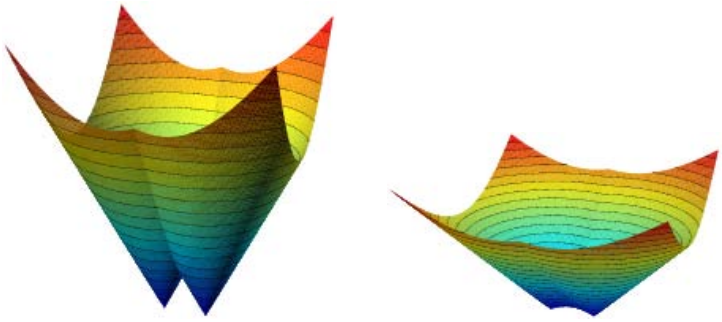


Figure 8.9: Non-convex interface: before reinitialization (left), after reintialization (right).

h	Direct reinit.		H-J reinit.		Proj. reinit.	
	Error	Order	Error	Order	Error	Order
0.05	0.08566	1.857	0.08948	1.345	0.00769	2.099
0.025	0.02306	1.936	0.03325	1.256	0.00183	2.260
0.0125	0.00595	1.952	0.01323	1.355	0.00041	2.035
0.0625	0.00152		0.00488		0.00010	

Table 8.1: Spatial convergence test: relative error on area $E = (A(\phi_h) - A_{\text{ex}})/A_{\text{ex}}$ for the three reinitialization procedures, circular interface.

h	Direct reinit.		H-J reinit.		Proj. reinit.	
	Error	Order	Error	Order	Error	Order
0.05	0.00387	1.421	0.01778	0.329	0.00283	1.517
0.025	0.00136	1.893	0.02698	1.041	0.00093	2.645
0.0125	0.00036	2.225	0.01295	1.268	0.00017	2.827
0.0625	0.00008		0.00511		0.00003	

Table 8.2: Spatial convergence test: relative error on area $E = (A(\phi_h) - A_{\text{ex}})/A_{\text{ex}}$ for the three reinitialization procedures, square interface.

h	Direct reinit.		H-J reinit.		Proj. reinit.	
	Error	Order	Error	Order	Error	Order
0.05	0.00830	2.16	0.03604	0.704	0.00544	2.144
0.025	0.00191	2.42	0.02561	1.090	0.00126	2.708
0.0125	0.00039	1.90	0.01175	1.233	0.00023	2.462
0.0625	0.00010		0.00476		0.00004	

Table 8.3: Spatial convergence test: relative error on area $E = (A(\phi_h) - A_{\text{ex}})/A_{\text{ex}}$ for the three reinitialization procedures, non-convex interface.

8.3.2 Effect of reinitialization on two-fluid flow problems

The need of an accurate and efficient reinitialization procedure can be better appreciated when observing its influence on the simulation of two-fluid flows. In this section, we present two examples of this kind of problems and we analyse the behaviour of different reinitialization techniques.

The first example concerns a gas bubble rising in a liquid. We refer to Chapter 9, where a wide range of bubble simulation is considered and analysed, for a complete description of the problem. At this stage, we are only interested in the effect of reinitialization. In Fig. 8.10, we present the contours of the level set function at t=0 s and t=0.1 s obtained without reinitialization and with different reinitialization techniques. Note that when the level set function is not reinitialized, a flat region is originated at the bottom side of the bubble and a very steep region is created in the front, with the obvious negative consequences in terms of accuracy. The direct and interface local projection reinitialization generate very similar contour fields. However, analysing the relative

error on the area inside the bubble at t=0.1 s, obtained on different grids (with h=1/10, 1/20, 1/40), we note the better behaviour of the proposed reinitialization (see Table 8.4). As in the mass conservation test case, the convergence order of the two methods is comparable and the error given by the interface local projection reinitialization in consistently lower than the one given by the direct reinitialization.

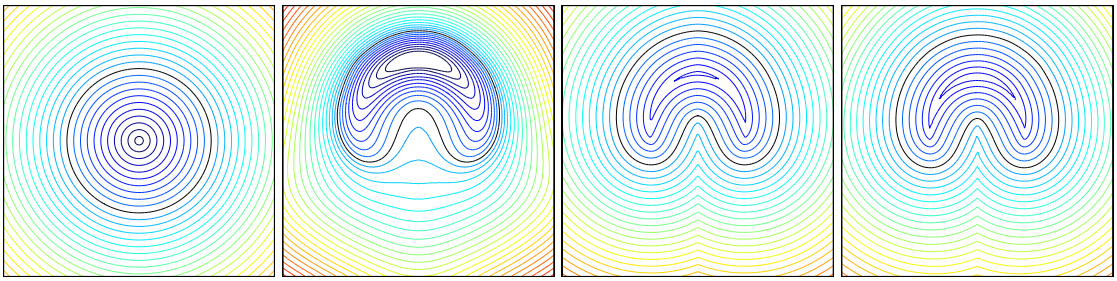


Figure 8.10: Contours of the level set function for the rising bubble problem, from left to right: initial solution, solution at time $t = 0.1$ s without reinitialization, solution at time $t = 0.1$ s with direct reinitialization, solution at time $t = 0.1$ s with interface local projection reinitialization.

h	Proj. Reinit.	Direct Reinit.
0.1	0.80 %	8.56 %
0.05	0.17 %	3.03 %
0.025	0.06 %	1.06 %

Table 8.4: Error on the area inside the bubble at time $t = 0.1$ s computed with respect to the initial area, for the interface local projection and the direct reinitializations.

Finally, we consider the Rayleigh–Taylor instability problem that will be described in detail in Section 9.3. A layer of heavier fluid, under the effect of gravity, penetrates into a lighter fluid giving rise to the typical *mushroom* shape. In Fig. 8.11, the contours of level set function associated to this problem is displayed. The initial level set function is defined as the signed distance from the curve $y(x) = 2 + 0.05\cos(2\pi x)$. When no reinitialization is used, we can note, as in the previous test case, the clustering of contour levels in the front of the heavier fluid, and a corresponding flattening behind the front. Direct and interface local projection reinitializations give similar results, although the direct reinitialization seems to be more dissipative in the high curvature regions.

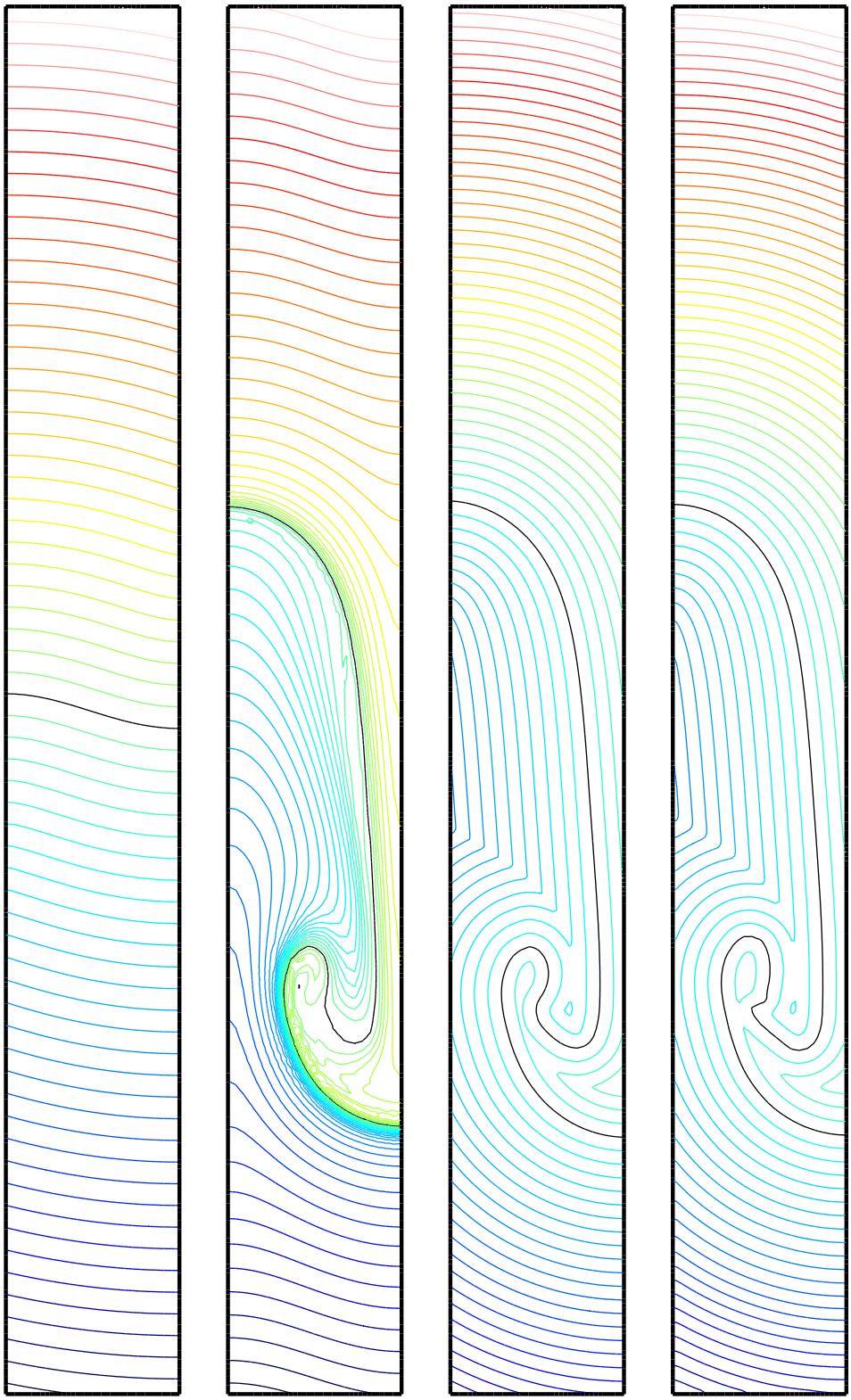


Figure 8.11: Effect of reinitialization for the Rayleigh–Taylor instability test case, from left to right: (*a*) initial level set function; (*b*) level set function at time $t = 3$s without reinitialization; (*c*) with direct reinitialization; (*d*) with interface local projection reinitialization. The black contour indicates the zero level set.

Chapter 9

Numerical Results of Two-Fluid Flow Simulations

In this chapter, we will present a set of numerical simulations carried out using the level-set finite element method introduced in the previous chapters. The different test cases that will be considered have been designed in order to assess the mathematical properties (stability, accuracy, robustness) of the method when dealing with a wide range of free-surface problems.

9.1 Bubble deformation under surface tension action

As a first test case, we consider a bubble in the absence of gravity with a square initial shape. Due to surface tension, the bubble shape evolves into a circle. For this simulation we have considered $\rho_1 = 1000$, $\rho_2 = 1$, $\mu_1 = 1$, $\mu_2 = 1$, and the surface tension coefficient $\sigma = 100$ N/m. The computational domain is the unit square and a uniform unstructured grid with $h = 1/40$ has been used. The time step is $\Delta t = 0.004$ s.

The shape evolution from square to circular is displayed in Fig. 9.1. The effect of the surface tension is initially localized in the corner zones (where the curvature attains its maximum values). The action of the surface tension can be better appreciated observing the velocity fields at different time instants (see Fig. 9.2). Note the peaks of velocity in the corner regions during the start up phase. As expected, the velocity magnitude decreases with time: the maximum velocity value at time $t = 0.033$ s is $\|\boldsymbol{u}_h\|_\infty = 1.68$ m/s, at time $t = 0.33$ s $\|\boldsymbol{u}_h\|_\infty = 0.68$ m/s and at time $t = 1$ s $\|\boldsymbol{u}_h\|_\infty = 0.28$ m/s.

We consider now an elliptic bubble in the absence of gravity. The initial bubble shape is defined by the ellipse:

$$\frac{(x-0.5)^2}{(a/2)^2} + \frac{(y-0.5)^2}{a^2} = 1,$$

where $a = 0.25$ is the y-semiaxis and the x-semiaxis is $a/2$. We consider the same physical properties as in the previous test case.

Under the effect of gravity the bubble undergoes a damped oscillation around the circular equilibrium configuration. The interface shape at different time instants is displayed in Fig. 9.3.

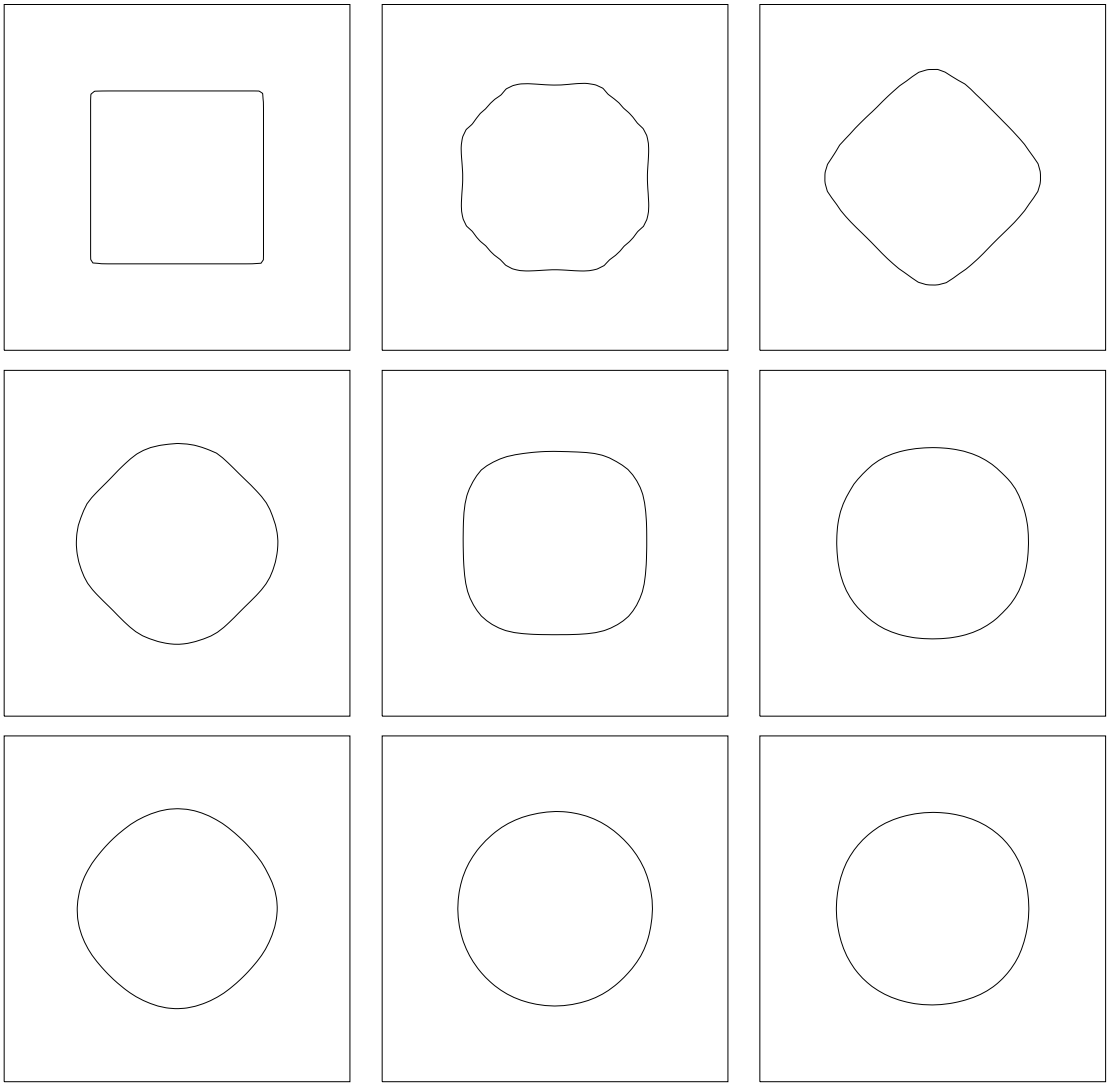


Figure 9.1: Square bubble evolving into a circular shape under the action of surface tension; interface position at time instants (from top right) $t = (0.1n)$ s (with $n = 0, \ldots, 8$).

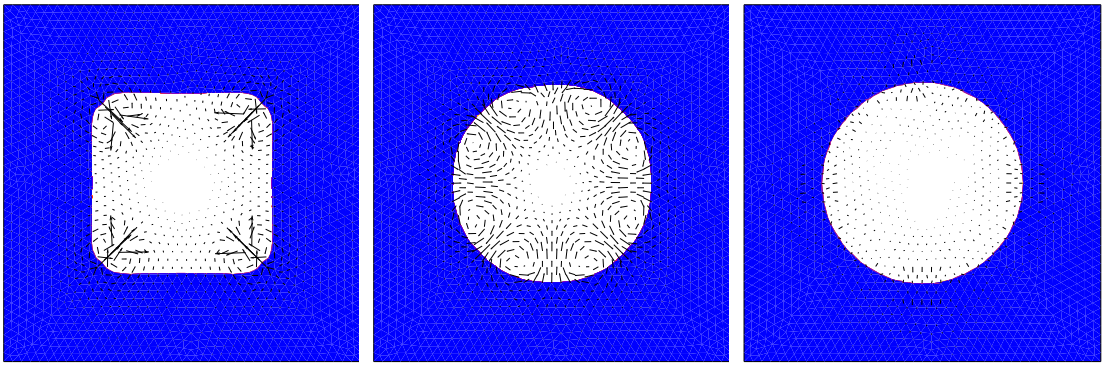


Figure 9.2: Interface position and velocity field for the square bubble test case at time instants $t = 0.033, 0.33, 1$ s.

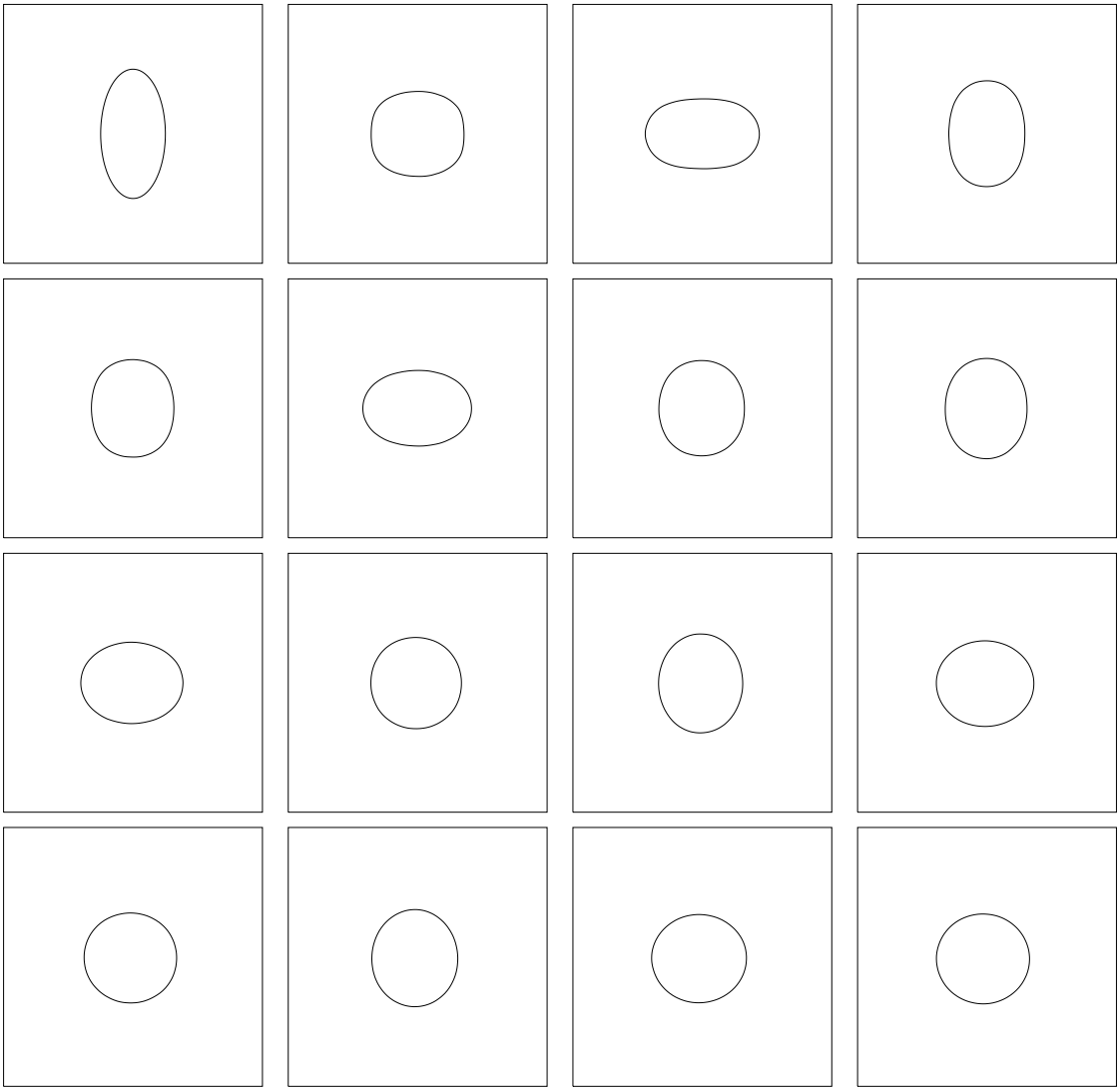


Figure 9.3: Elliptic bubble evolving into a circular shape under the action of surface tension, interface position at time instants (from top right) $t = (0.2n)$ s (with $n = 0, \ldots, 15$).

The results obtained can be compared with the analytical solution proposed in [Lam93]. In particular, the analytical expression of the oscillation frequency of a two-dimensional bubble of radius R in a liquid with density ρ is given by:

$$\omega_a = \left(\frac{6\,\sigma}{\rho\,R^3} \right)^{0.5}, \tag{9.1}$$

where σ is the surface tension coefficient. For the present case, expression (9.1) gives a theoretical oscillation frequency $\omega_a = 10.4$. In Fig.9.4 we report the time evolution of the bubble y-semiaxis obtained in our simulation using an unstructured grid with $h = 1/80$ and time step $\Delta t = 0.001$. The frequency of the bubble damped oscillation has been computed and is in good agreement with the analytical prediction. The discrepancy between the analytical and numerical values is about 12% for a coarse grid ($h = 1/40$), 5.6% for a medium size grid ($h = 1/80$) and 3.4% for a fine grid ($h = 1/160$).

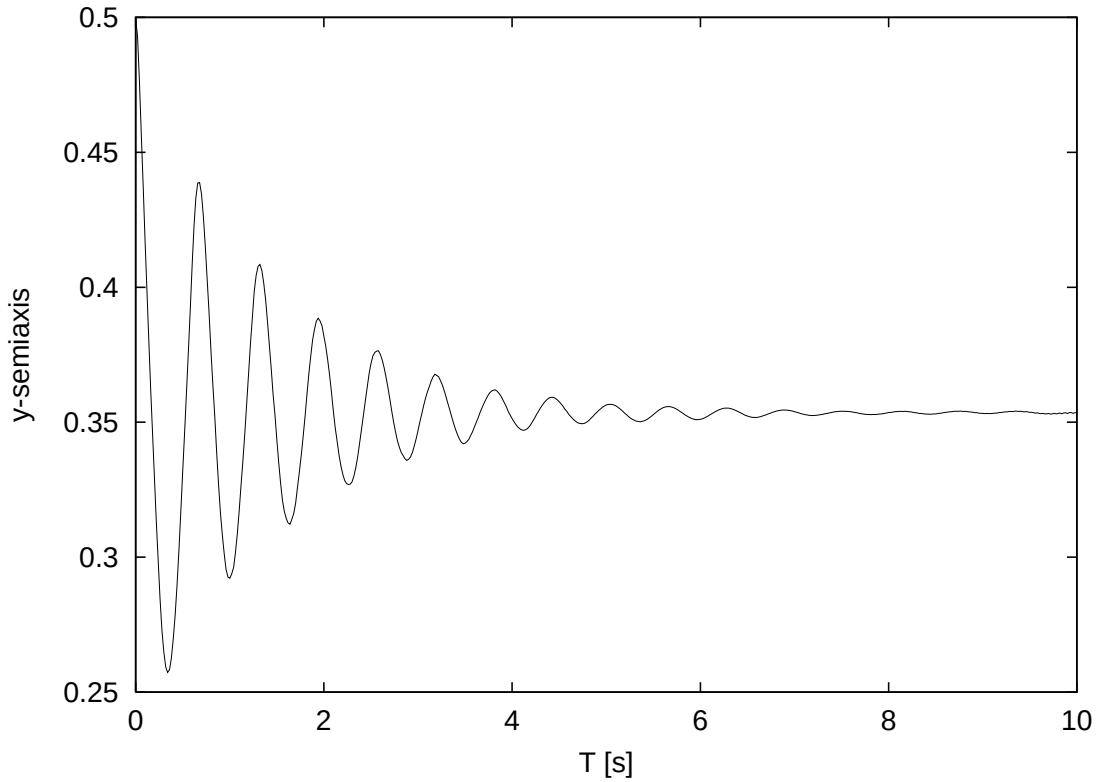


Figure 9.4: Time evolution of the y semiaxis for the elliptic bubble test case ($h = 1/80, \Delta t = 0.001$).

9.2 Rising Bubble

One of the classical problems used to validate numerical methods for two-fluid flows is the rising of a bubble of light fluid (typically a gas) into a heavier fluid (typically a liquid) due to buoyancy. The problem is characterized by four non-dimensional numbers. Two are the ratios of the bubble density and viscosity to the ones of the outer fluid. The remaining two numbers can be selected in a number of ways (see, *e.g.*, [ET98, Smo01, Tor00, vdPSV03, Cab03] for different choices of parameters). In the present work, we consider the Eötvös number, defined as:

$$\mathrm{Eo} = \frac{\rho_1 \, g \, D^2}{\sigma}, \tag{9.2}$$

where ρ_1 is the density of the outer fluid, D is the diameter of the bubble and σ is the surface tension coefficient. The Eötvös number measures the relative importance of buoyancy versus surface tension forces. The last non-dimensional number is the Galileo (or Archimedes) number (see [CGW78]), defined as:

$$\mathrm{N} = \frac{\rho_1^2 \, D^3 \, g}{\mu_1^2}, \tag{9.3}$$

which is a Reynolds number squared based on the velocity scale $\sqrt{D\,g}$, and indicates the ratio of buoyancy to viscous forces. In this section, we consider a set of rising bubble simulations over a range of Eötvös and Galileo numbers.

We first consider a low Galileo number case. The parameters are the same as in [ET98] and [dSMN$^+$04] and are given by Eo=1, N= $10^{3/2}$, $\rho_1/\rho_2 = 20$ and $\mu_1/\mu_2 = 20$. These parameters correspond to an air bubble of diameter 1.9 mm rising in standard oil ($\rho = 880$ kg/m^3, $\mu = 0.21$ Ns/m^2 and $\sigma = 0.03$ N/m). The computational domain is rectangular with dimensions $L \times 2L$, with $L = 120$ mm. No-slip boundary conditions are imposed on the top and bottom sides, while on the lateral side we consider free-slip boundary conditions.

In [ET98], results are given in terms of bubble rising velocity and centroid position. The bubble rising velocity is defined by:

$$v_b = \int_b \boldsymbol{u} \cdot \boldsymbol{j} \, d\boldsymbol{x},$$

where the integral is computed in the interior of the bubble and $\boldsymbol{j}$ is the unit vector in vertical direction. Analogously, the coordinates of the bubble centroid $\boldsymbol{x} = (x_b, y_b)$ are given by:

$$\boldsymbol{x}_b = \int_b \boldsymbol{x} \, d\boldsymbol{x}.$$

We have considered four uniform unstructured grids with $h = L/15, L/30, L/60, L/120$ mm and a time step $\Delta t = 10^{-4}$ s. The bubble interfaces and the velocity fields at $t = 0.1$ computed on different grids are shown in Fig. 9.5. Note the formation of the typical couple of counterrotating vortices associated to the rising of the bubble.

The evolution of Reynolds number based on the rising velocity with respect to non-dimensional time ($T = t/\sqrt{gD}$) is presented in Fig. 9.6. For the two coarser grids, oscillations on the rising velocity are observed. These oscillations disappear when the grid is refined and the rising velocity

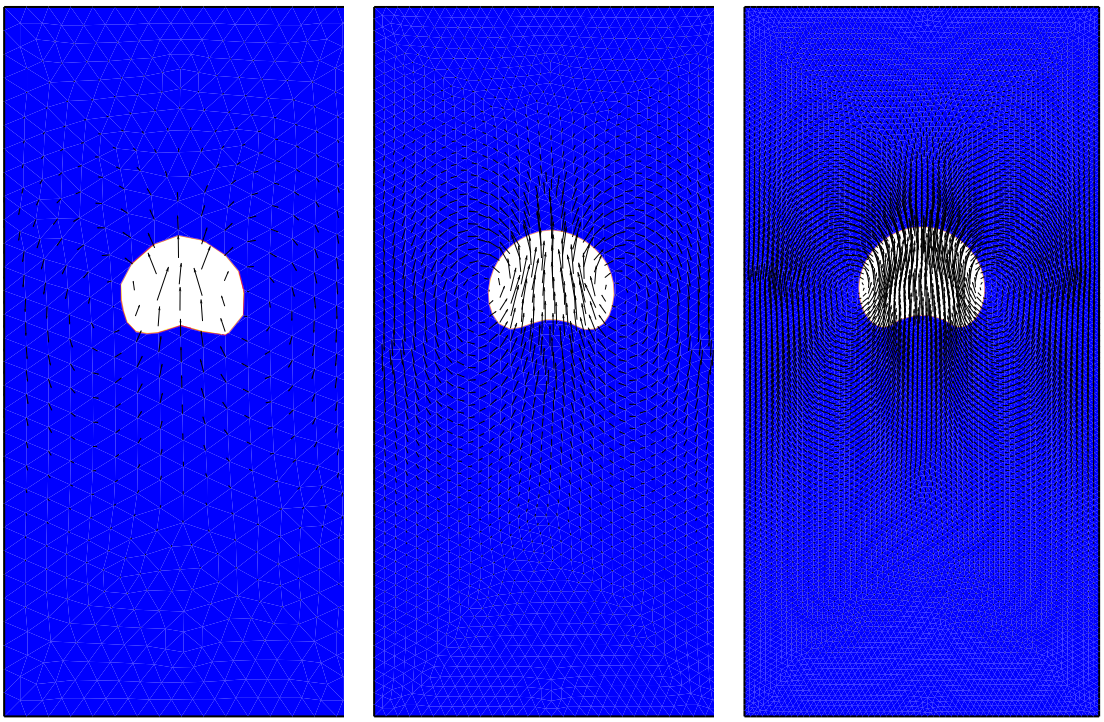


Figure 9.5: Bubble interface and the velocity field at $t = 0.1$ on three grids: $h = 1/15$ (left), $h = 1/30$ (center) and $h = 1/60$ (right).

profile for the two finer grids are not distinguishable any longer. The asymptotic rising Reynolds number is about 1.5, in good agreement with results obtained in [ET98] and [dSMN+04]. The time evolution of the non-dimensional vertical coordinate of the bubble centroid ($Y_b = y_b/D$) is presented if Fig. 9.7. Again, grid convergence is observed as the grid is refined.

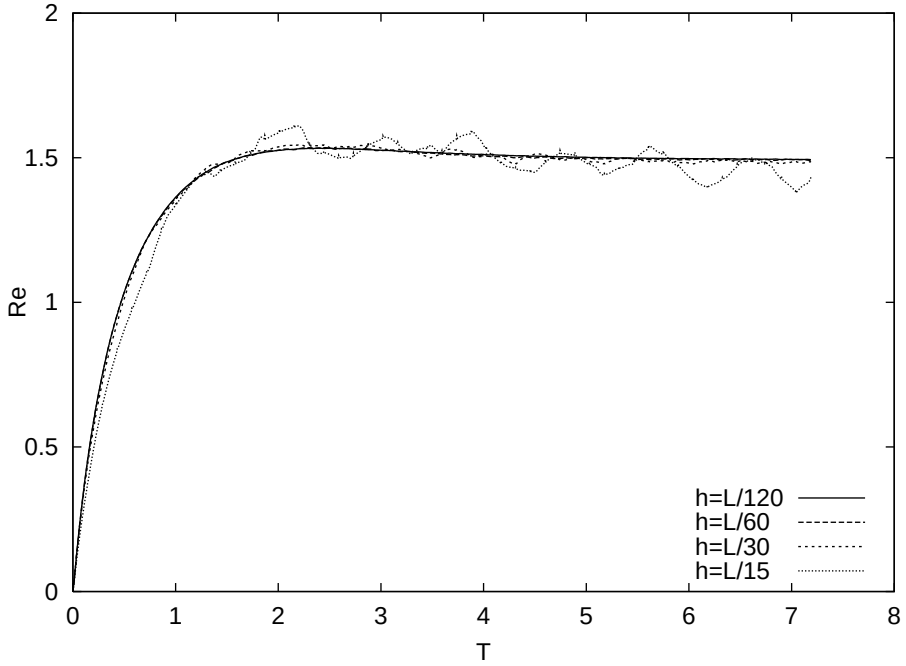


Figure 9.6: Rising Reynolds number versus non-dimensional time on four grids ($h = 1/15, 1/30, 1/60, 1/120$).

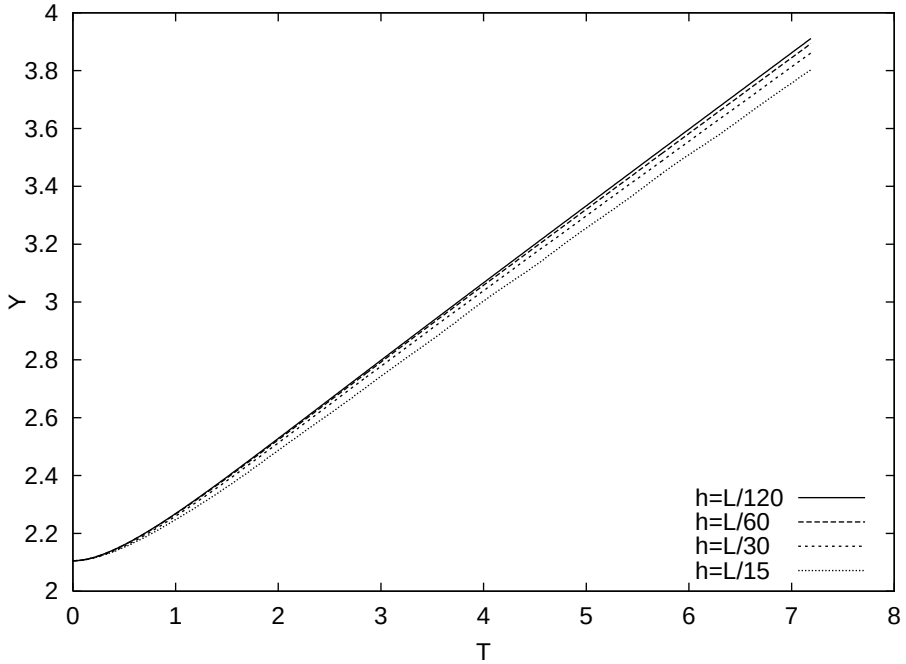


Figure 9.7: Non-dimensional vertical coordinate of bubble centroid versus non-dimensional time on four grids ($h = 1/15, 1/30, 1/60, 1/120$).

The asymptotic shape of a rising bubble, as well as its rising velocity, depend on the four non-dimensional parameters introduced above. Several experimental and numerical investigations concerning the different rising regimes can be found in the literature (see, *e.g.*, [CGW78, HB76] as experimental references and [Smo01, CGRL99, SS97] as numerical references in two and three dimensions). In particular, the book by Clift *et al.* [CGW78] reports a diagram of the qualitative dependence of the bubble shape on Eötvös and Reynolds numbers.

We have performed a set of numerical simulations for different values of the Galileo (and, therefore, Reynolds) number, in order to analyse its effect on the bubble rise. The computational domain and the physical parameters are identical to the ones considered in the rising velocity study presented above, namely Eo=1, $\rho_1/\rho_2 = 20$, $\mu_1/\mu_2 = 20$. Four values of Galileo numbers have been considered: N=1, N=100, N=4900 and N=10000, which correspond to Reynolds numbers Re=1, Re=10, Re=70 and Re=100, respectively. Based on the spatial convergence results obtained above, a computational grid with $h = L/30$ has been used. The time step is $\Delta t = 10^{-4}$. The bubble is initially circular and the initial velocity is null everywhere.

The evolution of the bubble interface for different values of Galileo number is displayed in Fig. 9.8. We can observe that, in accordance to the mentioned qualitative diagram presented in [CGW78], the asymptotic rising shape evolves from a quasi-circular shape (for low values of N), to a *dimpled* ellipsoidal cap shape, when N is increased. Note that, for the highest values of N considered, the bubble become skirted, and eventually the skirts break offs due to the action of the vortices in the bubble wake. The remaining part of the bubble develops a spherical cap shape. The small bubbles generated by the breaking rise with a lower velocity than the original bubble, due to their smaller volume. It is quite remarkable that, even if the computational grid covers these small bubbles with just few elements, their volume seems to be reasonably conserved. On the other hand, a better resolution of the dynamics of these small bubbles would require a finer grid. This result confirms the satisfactory mass conservation properties of our method, owing to the new stabilization technique and reinitialization procedure introduced and discussed in the previous chapters.

Also the rising velocity depends on the Galileo number. In particular, it has been shown in [CGRL99] that rising velocity increases with N. We have investigated the effect of different rising velocities, together with different bubble shapes, on the development of the wake beneath the bubble. In Fig. 9.9, we present the streamlines in the local reference frame (attached to the rising bubble) around and inside the bubble, for different values of Galileo number. For the case with N=1 (Re=1), the flow is completely attached and we can note the two counterrotating vortices inside the bubble. Increasing the Galileo number to 100 (Re=10) produces a deformation of the bubble shape, but the flow remains attached and no wake is produced. For N=4900 (Re=70), we note the formation of a recirculation region behind the bubble. When N is further increased, the recirculation region becomes longer and asymmetric and eventually becomes unstable. In fact, these results reproduce the well known behaviour of the flow past a blunt body for different Reynolds numbers. Moreover, our results are in good agreement with the numerical results presented in [UT92] and with the experiments discussed in [BW81].

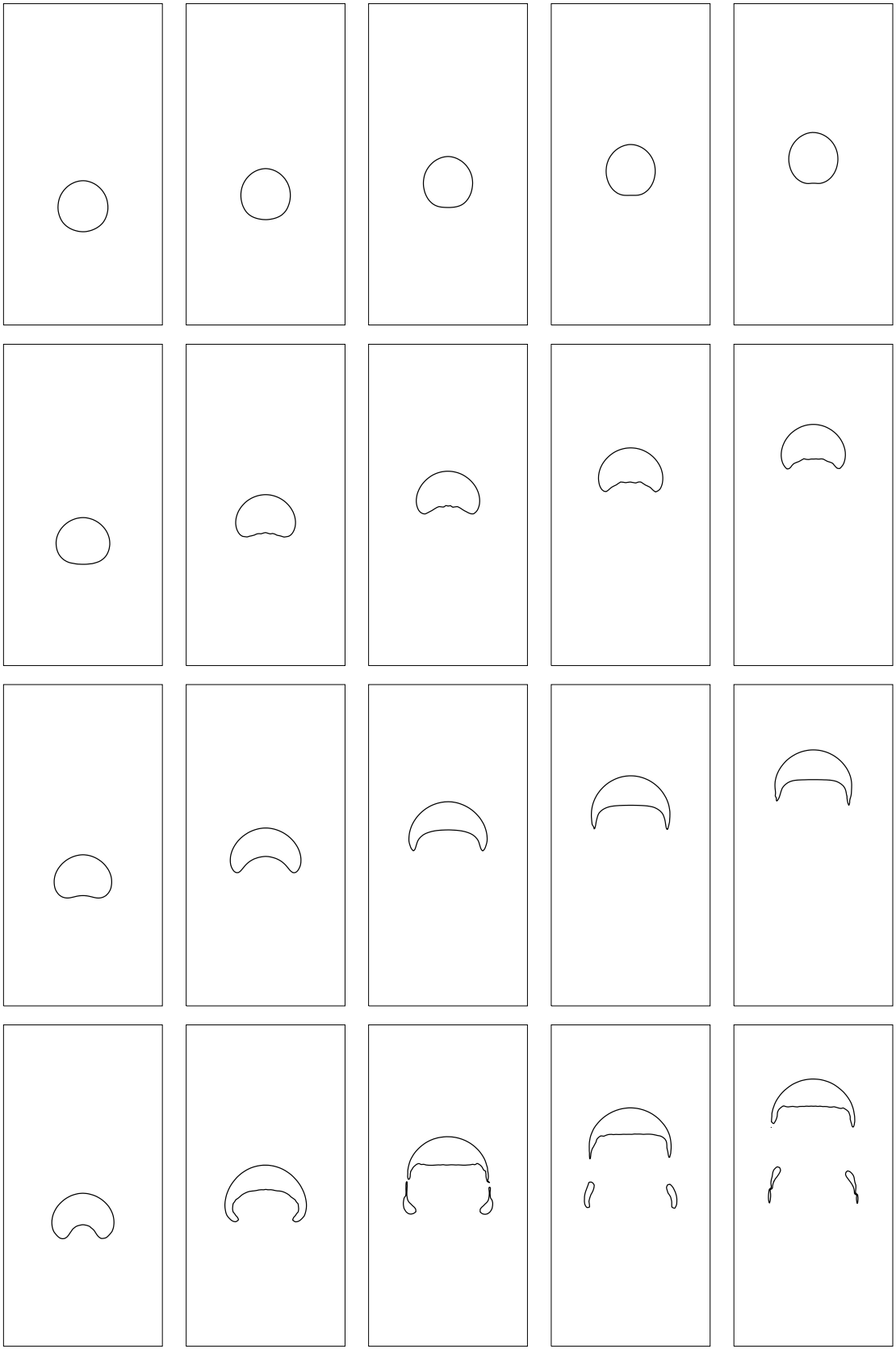


Figure 9.8: Influence of Galileo number on bubble rise. From top to bottom: N=1 (Re=1), N=100 (Re=10), N=4900 (Re=70), N=10000 (Re=100). Time instants, from left to right: t=0.02, 0.04 0.06, 0.08, 0.1 s.

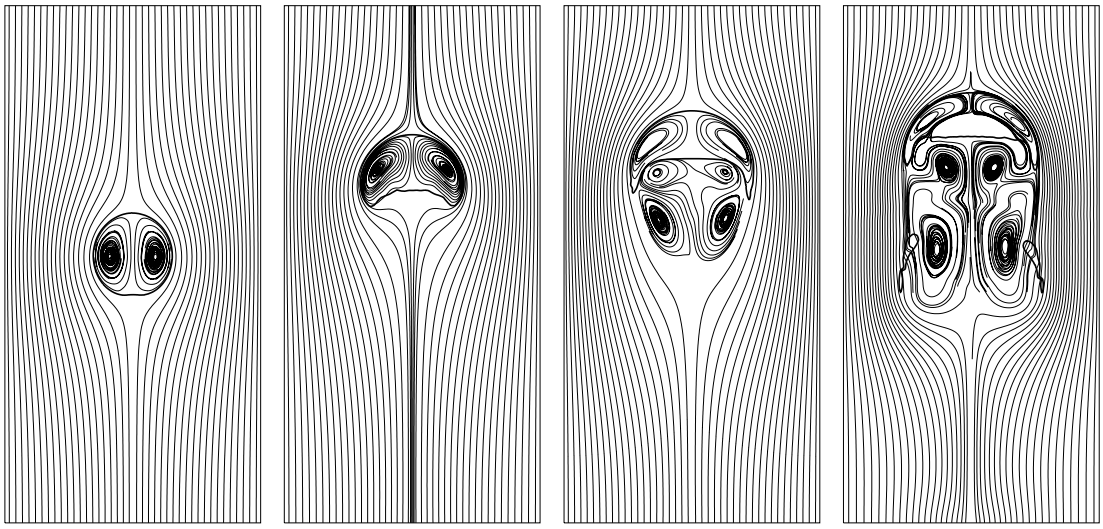


Figure 9.9: Interface position and streamlines in the local reference frame for different values of the Galileo number, from left to right: N=1 (Re=1), N=100 (Re=10), N=4900 (Re=70), N=10000 (Re=100)

9.3 Rayleigh-Taylor instability

The Rayleigh–Taylor instability is a well known phenomenon occurring in two-fluid flows, characterized by the acceleration of a heavier fluid into a lighter one under the action of gravity. In the literature, several numerical investigations of this phenomenon can be found (see, *e.g.* [BM80, TU90, BM92, PZ99]).

We consider the same simulation conditions as in [BM92, Smo01]: the computational domain is a box with sizes $[0, 0.5]$ m $\times$ $[0, 4]$ m, the viscosities of the two fluids are $\mu_1 = \mu_2 = 0.003$ kg/(m s), the fluid densities are $\rho_1 = 0.17$ kg/m^3, the gravity acceleration is $g = 1.0$ m/s^2 the surface tension coefficient is zero. The level-set is initialized as the signed distance function from the initial interface given by the function $y(x) = 2 + 0.05\cos(2\pi x)$. The fluid is initially at rest and no-slip wall condition is imposed on the top and bottom sides, while a symmetry boundary condition is imposed on the lateral sides. We consider a uniform unstructured triangulation with $h = 1/40$ and time step $\Delta t = 0.01$.

The Rayleigh–Taylor instability is activated by the initial sinusoidal perturbation. In Fig. 9.10, we can observe how the heavy flow start penetrating the lighter one and rolling up along the side of the spike, generating the typical *mushroom* shape. The interface evolution is in good agreement with analogous results found in the literature [PZ99, Smo01]. To highlight the stabilizing effect given by the surface tension, the same simulation has been performed considering the surface tension coefficient $\sigma = 0.015$ N/m. In Fig. 9.11, we can observe the effect of the surface tension in delaying the development of the instability.

Finally, we have carried out the Rayleigh–Taylor instability test case (without surface tension) using the Volume of Fluid method described in Section 1.2.2 and employed in the numerical simulation presented in Chapter 4 anf 5. In Fig. 9.12, the evolution of the interface computed by the Volume of Fluid method with the same discretization parameters considered above ($h = 1/40$ m, $\Delta t = 0.01$ s) is presented. The limitations of the VOF approach in terms of interface artificial diffusion is evident.

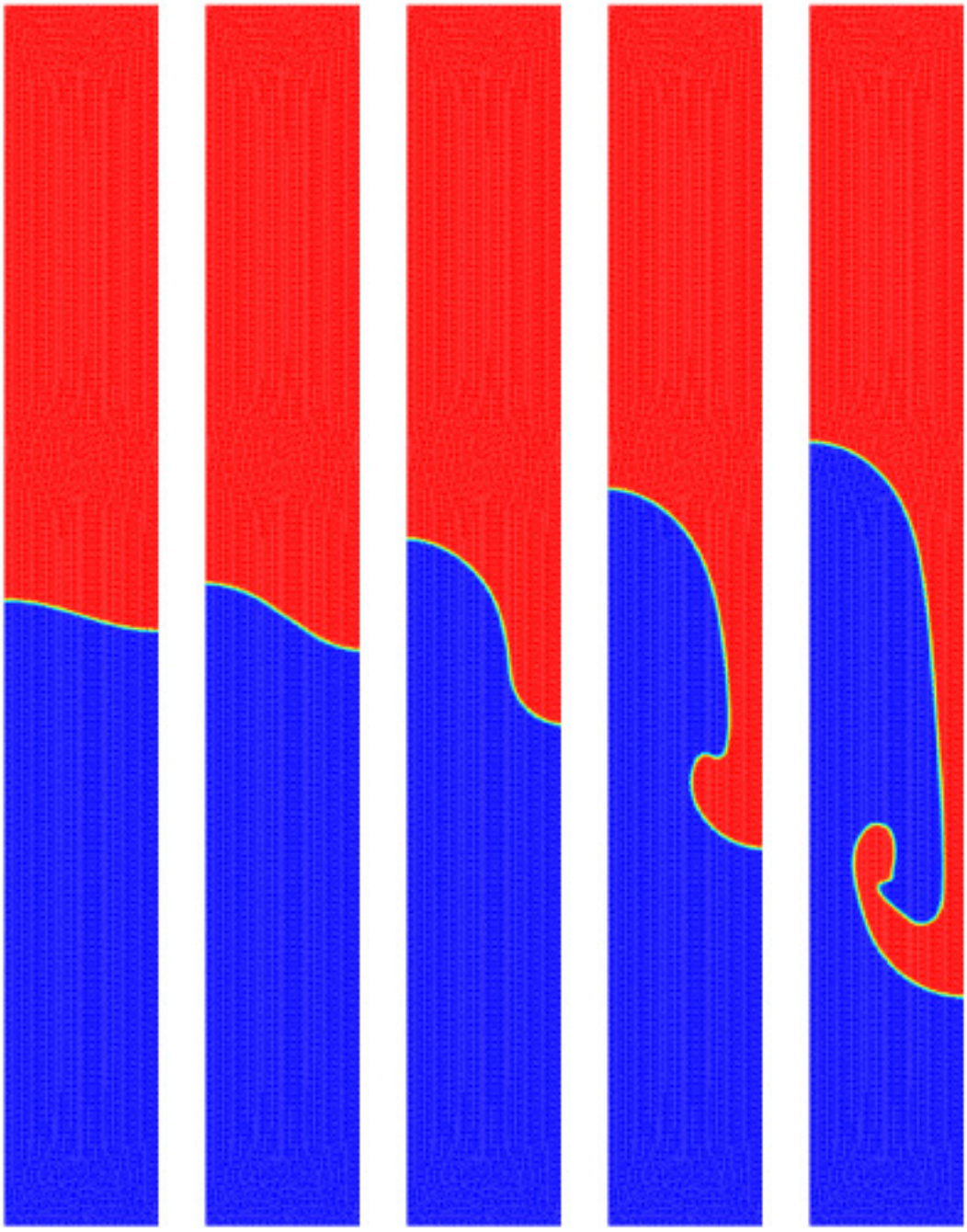


Figure 9.10: Rayleigh–Taylor instability without surface tension ($h = 1/40$ m, $\Delta t = 0.01$ s): density distribution at time $t = 0, 0.75, 1.5, 2.25, 3$ s.

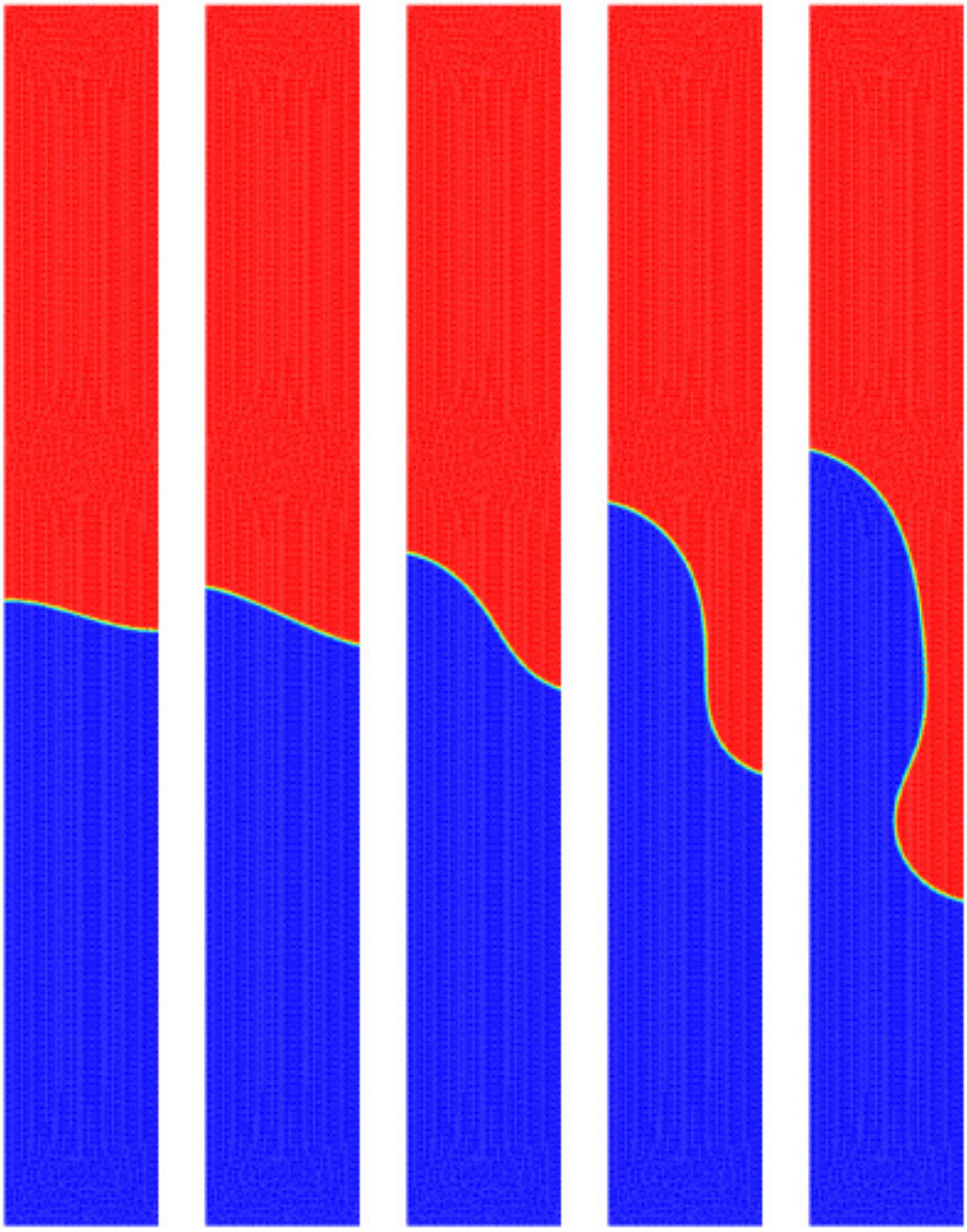


Figure 9.11: Rayleigh–Taylor instability with surface tension ($\sigma = 0.015$, $h = 1/40$ m, $\Delta t = 0.01$ s): density distribution at time $t = 0, 0.75, 1.5, 2.25, 3$ s.

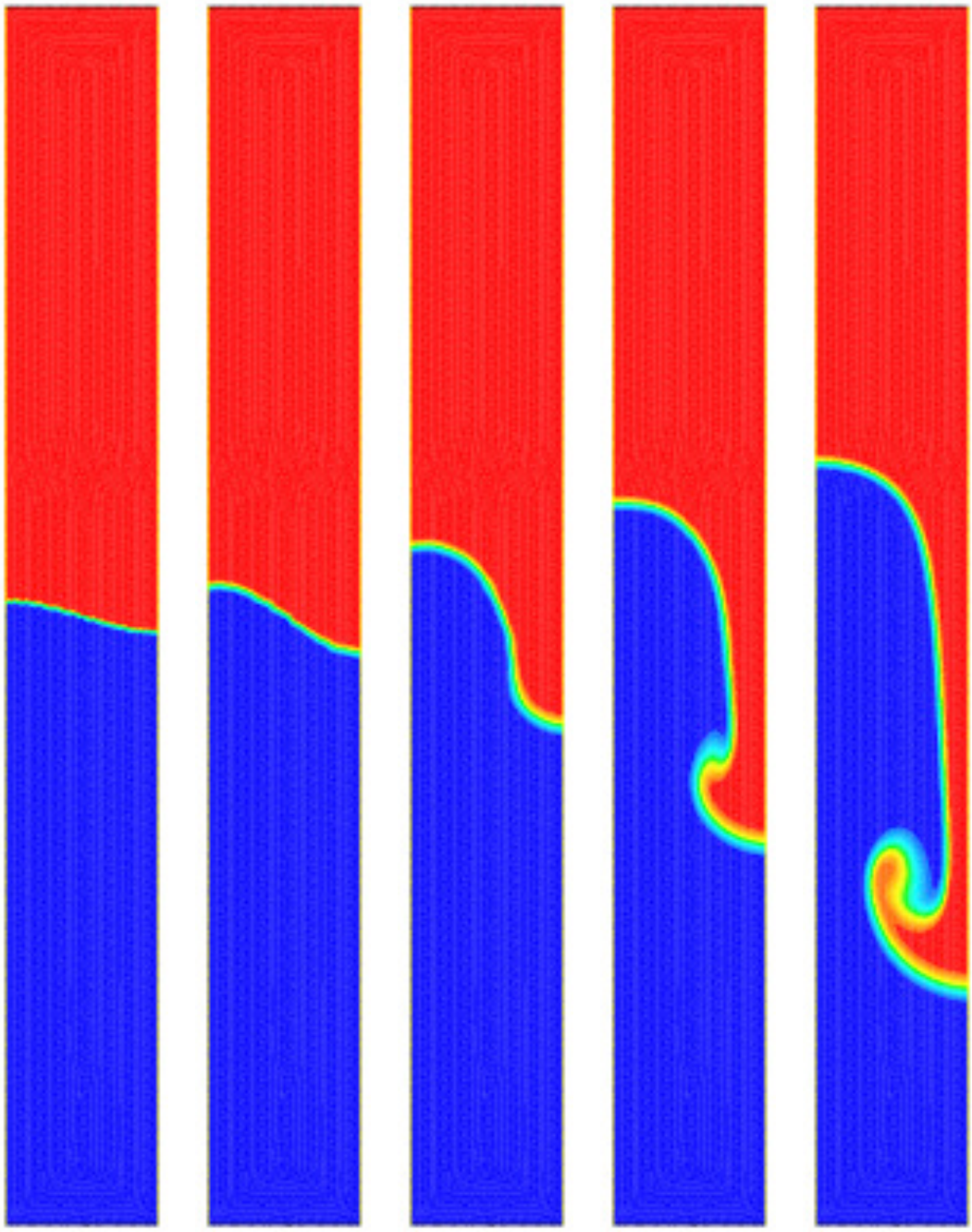


Figure 9.12: Rayleigh–Taylor instability without surface tension computed with a Volume of Fluid method ($h = 1/40$ m, $\Delta t = 0.01$ s): density distribution at time $t = 0, 0.75, 1.5, 2.25, 3$ s.

9.4 Drop falling into a free-surface

The following numerical example has been designed to show the ability of the proposed method to deal with topology changes in two fluid flows. A liquid droplet is left falling through the air into a layer of liquid at rest. The computational domain is the unit square and the droplet is left falling from a distance equal to the initial radius of the droplet. For this example, we have considered $\rho_2/\rho_1 = 1000$, $\mu_2/\mu_1 = 1$, Re=100 and the surface tension coefficient equal to zero. A uniform unstructured grid with $h = 0.0125$ and a time step $\Delta t = 0.002$ s have been used. The evolution of the interface, with the change of topology occurring when the droplet touches the free-surface, is displayed in Fig. 9.13, together with the corresponding velocity field. We note that two counterrotating vortices are generated during the droplet fall and the diverging vortices are associated to surface waves resulting from the water splash.

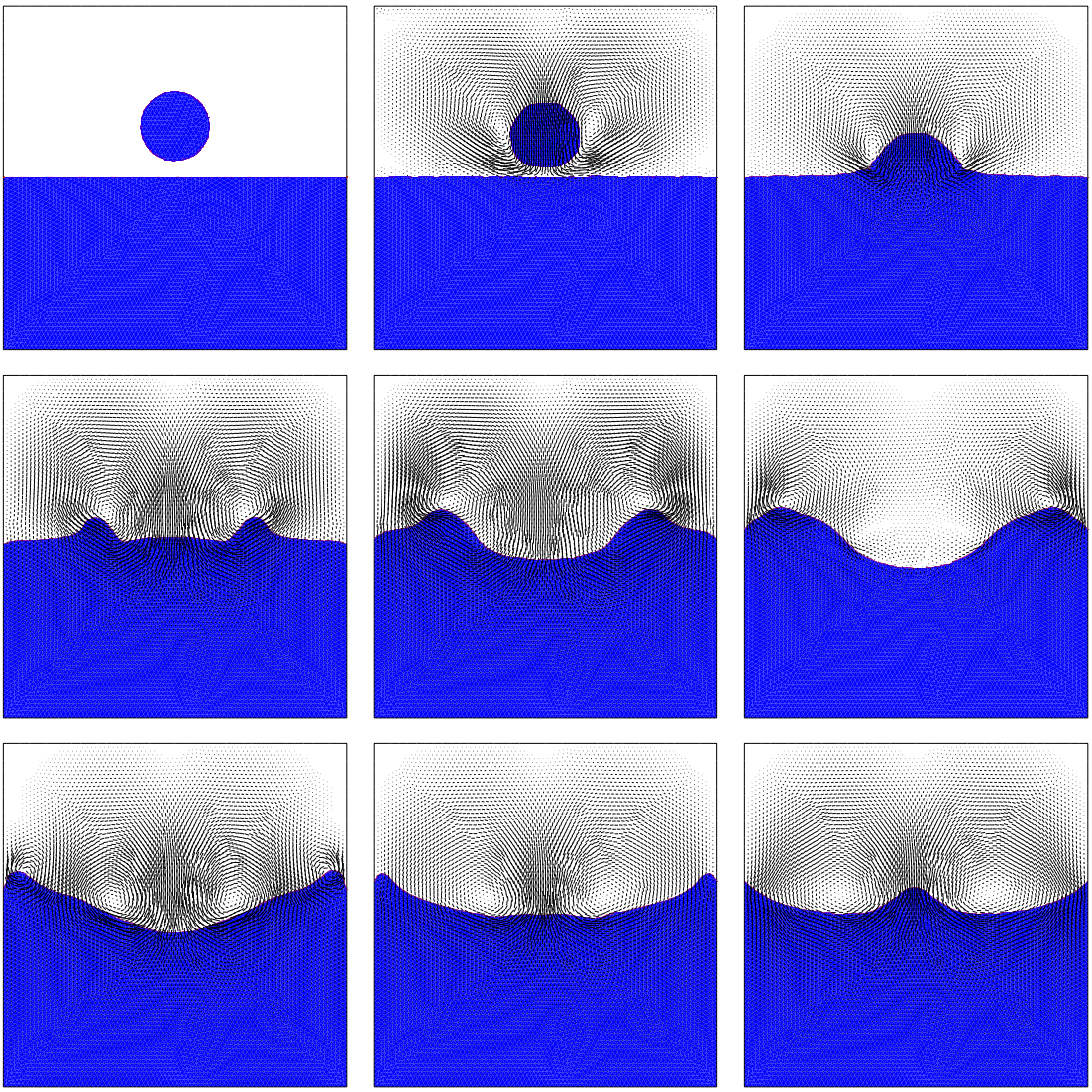


Figure 9.13: Drop falling into a free-surface (Re=100, $\rho_2/\rho_1 = 1000$, $\mu_2/\mu_1 = 1$): interface position and velocity field at time instants $t = 0, 0.08, 0.16, 0.24, 0.32, 0.40, 0.48, 0.56, 0.64$ s.

9.5 Broken dam problem

The broken dam problem is a classical test case used to validate free-surface numerical models (see, textite.g., [MPR99, IOnDP04]). The reference experimental data are taken from [MM52]. A sketch of the experimental setup is reported in Fig. 9.14. In our simulations, we consider a square column of liquid of size $H = 0.05715$ m in hydrostatic equilibrium that is allowed to collapse under the effect of gravity in a 0.17145 m $\times$ 0.07 m rectangular cavity, as the right-hand side wall of the liquid column is removed. The density and viscosity ratios are $\rho_1/\rho_2 = 1000$ and $\mu_1/\mu_2 = 100$. Free-slip conditions are imposed at the boundary of the rectangular cavity.

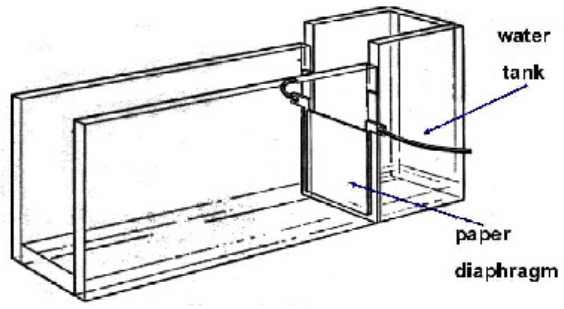


Figure 9.14: Experimental setup used in [MM52].

Three uniform triangular grids obtained from, respectively, a 20×40, 40×80 and 80×160 cartesian grid have been considered, with a time step $\Delta t = 10^{-3}$ s. The evolution of the free-surface is displayed in Fig. 9.15 together with the velocity field for the 40×80 grid.

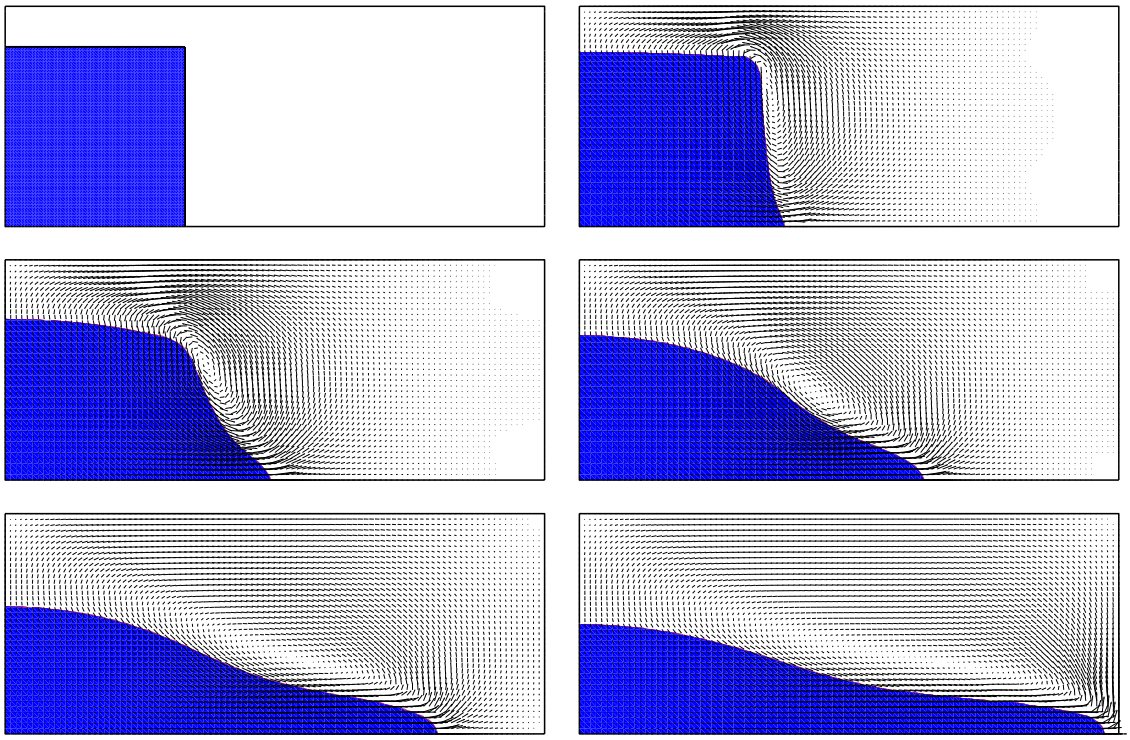


Figure 9.15: Interface location and vector field at t =0 s, 0.2 s, 0.4 s, 0.6 s , 0.8 s and 1 s.

Fig. 9.16 shows the time evolution of the interface front on the bottom side obtained from the simulations on the three grids compared with the experimental results reported in [MM52]. For convenience, in Fig. 9.16 non-dimensional space and time, defined as $T = t\sqrt{g/H}$ and $X = x/H$, are considered.

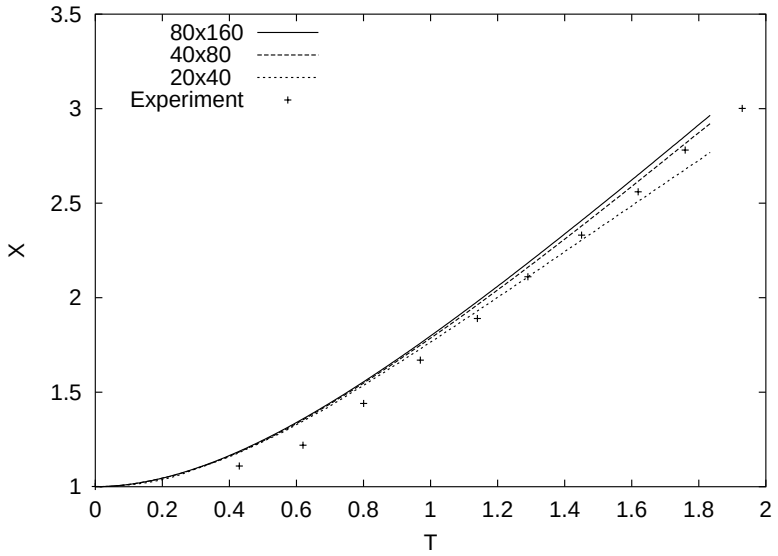


Figure 9.16: Comparison between the time evolutions of the interface front computed on the three grids and the experimental data.

The numerical solutions are in reasonable agreement with the experimental results. The discrepancy between numerical and experimental results can be attributed to the lack of three-dimensional effects in the 2d simulations as well as to the use of a non-physical free-slip condition.

Conclusions

In this thesis, we have addressed the numerical modelling of viscous free-surface flows. We have employed state-of-the-art numerical methods, based on the solution of the Navier–Stokes equations, for the simulation of flow problems in the domain of ship hydrodynamics.

In the first part, the role of Computational Fluid Dynamics in the design process of racing boats has been discussed. We have presented the numerical investigations on the flow around the appendages and the sails of an America's Cup yacht, that was carried out in the framework of the collaboration with the Swiss Team Alinghi. We have shown how the overall design process can benefit from an adequate numerical simulation procedure, based on an effective grid generation strategy, intensive numerical computations and post-processing efforts. In particular, we have analysed some of the parameters that define the geometry of the yacht appendages and we have obtained indications about the performances associated to the different configurations considered.

A similar approach has been adopted for the simulation of the flow around an Olympic rowing boat. In this case, we have focused our investigations on the free-surface generated by an advancing rowing hull. The numerical approach has been validated through comparisons with experimental results. Moreover, different hull designs have been analysed and we have obtained relative performance estimations. To take into account the effect of the complex dynamics of the hull during the race, we have set up a simplified approach based on the numerical simulation of the flow around the hull subjected to an imposed pitching motion. This analysis led us to propose an innovative design modification, consisting in the adoption of pitch stabilizers, that has been found to be beneficial on the overall performance of the boat. Due to the simple modelling of the fluid-structure interaction, further investigations on this subject would be required to fully assess this improvement.

A critical analysis of the results obtained for this class of flow problems has highlighted some limitations of the numerical model adopted. In particular, the accuracy in the prediction of the free-surface dynamics has been found not to be completely satisfactory.

In the second part of this work, we have addressed this aspect from a more theoretical and methodological point of view. A new numerical method for the solution of two-fluid flows has been introduced. This method is based on a level set approach implemented in a finite element framework. Up to now, the combination between level set method and finite element approximation has not been extensively explored in the literature. We have shown that, if the most critical aspects related to its implementation are adequately faced, an accurate and effective scheme can be obtained.

For the solution of the pure advection equation, which arises in the level set method, we have proposed a new stabilization method based on an interior penalty approach. We have introduced a local stabilization term penalizing the jump of the gradient and we have proved that this method

shares the stability and convergence properties of the subgrid viscosity stabilization, yielding a slightly less computationally expansive scheme. The advantages of this method when compared with classical stabilization techniques have been assessed through numerical tests.

To address the solution of the reinitialization problem, which is known to be a critical aspect in level set methods, we have proposed a strategy based on a local (in the interface region) reconstruction of the level set function, which gives the best (in the L^2 sense) piecewise linear approximation of the interface position. The convergence analysis that we have carried out shows that optimal error estimates for the reinitialization procedure can be obtained.

Finally, the proposed finite element level set method has been used for the simulation of a variety of test cases concerning laminar two-fluid flows. The results of these simulations have been presented and discussed. The method has been implemented in a finite element library which is restricted to two-dimensional problems. However, the methodology proposed is well suited for the solution of three-dimensional problems.

Bibliography

[ABC03] Y. Achdou, C. Bernardi, and F. Coquel, *A priori and a posteriori analysis of finite volume discretizations of Darcy's equations*, Numer. Math. **96** (2003), no. 1, 17–42.

[ACS03] R. Azcueta, M. Caponnetto, and H. Soeding, *Motion simulations for planing boats in waves*, Proc. of the 15th. International Conference on Hydrodynamics in Ship Design, Safety and Operation (Gdansk, Poland), 2003.

[Ale25] F. H. Alexander, *The theory of rowing*, Proceedings of the University of Durham Philosophical Society (Newcastle-upon-Tyne, England) (G. W. Todd, ed.), vol. VI, Andrew Reid & Co., 1925.

[AP91] N. Ashgriz and J. Y. Poo, *Flair: fluz line-segment model for advection and interface reconstruction*, J. Comp. Phys. **93** (1991), no. 2, 449–468.

[Azc01] R. Azcueta, *Computation of turbulent free-surface flows around ships and floating bodies*, Ph.D. thesis, Technical University of Hamburg-Harburg, 2001.

[Azc02a] _____, *Computation of turbulent free-surface flows around ships and floating bodies*, Ship Tech. Res. **49** (2002), –.

[Azc02b] _____, *RANSE simulations for sailing yachts including dynamic sinkage & trim and unsteady motions in waves*, Proc. of the High Performance Yacht Design Conference (Auckland), 2002, pp. 13–20.

[Bar89] T. J. Barth, *The design and application of upwind schemes on unstructured meshes*, AIAA Report **89-0366** (1989).

[BB04] R. Becker and M. Braack, *Numerical Mathematics and Advanced Applications*, ch. A two-level stabilization scheme for the Navier-Stokes equations, pp. 123–130, Springer, 2004.

[BdM96] M. Brearley and N. J. de Mestre, *Modelling the rowing stroke and increasing its efficiency*, Proceedings of the 3rd Conference on Mathematics and Computers in Sport (Quensland, Australia), 1996, pp. 35–46.

[BF91] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer-Verlag New York, Inc., 1991.

[BH82] A. N. Brooks and Hughes, *Streamline upwind / Petrov–Galerkin formulations for flows with particular emphasis on the incompressible Navier–Stokes equations*, Comp. Meth. Appl. Mech. Engng **32** (1982), 199–259.

[BJWT99] M. Beddhu, M.Y. Jiang, D.L. Whitfield, and L.K. Taylor, *Computation of the wetted transom stern flow over model 5415*, Proc. of the 7th International Conference on Numerical Ship Hydrodynamics (Nantes), 1999.

[BKZ92] J. U. Brackbill, D. B. Kothe, and C. Zemach, *A continuum method for modeling surface tension*, J. Comp. Phys. **100** (1992), no. 2, 335–354.

[BL78] B. S. Baldwin and H. Lomax, *Thin layer approximation and algebraic model for separated turbulent flows*, 16th Aerospace Sciences Meeting, Reno, 1978, AIAA paper 78-257.

[BM80] G.R. Baker and S.A Meiron, D.I.and Orszag, *Vortex simulations of the Rayleigh–Taylor instability*, Phys.Fluids **23** (1980), 1485–1490.

[BM92] J. B. Bell and D. L. Marcus, *A second-order projection method for variable-density flows*, J. Comput. Phys. **101** (1992), 334–348.

[Bou97] J. Boussinesq, *Théorie de l'écoulement tourbillonnant et tumultueux des liquides dans les lits rectilignes à grandes sections*, Gauthier-Villars, Paris, 1897.

[BP79] M. Bercovier and O. Pironneau, *Error estimates for finite element solution of the stokes problem in the primitive variables*, Numer. Math. **33** (1979), 211–224.

[BP04a] E. Burman and N. Parolini, *Local edge stabilization for the transport equation*, 2004, in preparation.

[BP04b] ———, *A new reinitialization procedure for the finite element approximation of the level set equation*, 2004, in preparation.

[Bre83] H. Brezis, *Analyse fonctionnelle*, Masson, Paris, 1983.

[BS98] T.J. Barth and J.A. Sethian, *Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains*, J. Comp. Phys. **145** (1998), no. 1, 1–40.

[BSHS99] F. Bet, N. Stuntz, D. Hanel, and S. Sharma, *Numerical simulation of ship flow restricted water*, Proceedings of the 7th International Conference on Numerical Ship Hydrodynamics (Nantes), 1999.

[Bur04] E. Burman, *A unified analysis for conforming and nonconforming stabilized finite element methods using interior penalty*, SIAM J. Numer. Anal. (2004), Accepted.

[BW81] D. Braga and M. E. Weber, *Bubbles in viscous liquids: shapes, wakes and velocities*, J. Fluid Mech. **105** (1981), 61–85.

[CA02] M. S. Celebi and H. Akyildiz, *Nonlinear modeling of liquid sloshing in a moving rectangular tanks*, Ocean Eng. **29** (2002), no. 12, 1527–1553.

[Cab03] A. Caboussat, *Analysis and numerical simulation of free surface flows*, Thesis n. 2893, École Polytechnique Fédérale de Lausanne (EPFL), 2003.

[CFN95] L. Corrias, M. Falcone, and R. Natalini, *Numerical schemes for conservation laws via Hamilton–Jacobi equations*, Math. Comp. **64** (1995), 555–580.

[CGRL99] L. Chen, L. S. V. Garimella, J. A. Reizes, and E. Leonardi, *The development of a bubble rising in a viscous liquid*, J. Fluid Mech **387** (1999), 61–96.

[CGW78] R. Clift, J. R. Grace, and M. E. Weber, *Bubbles, drops and particles*, Academic Press, 1978.

[CHMO96] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher, *A level set formulation of eulerian interface capturing methods for incompressible fluid flows*, J. Comp. Phys. **124** (1996), no. 2, 449–464.

[Cho68] A. J. Chorin, *Numerical solution of the Navier–Stokes equations*, Math. Comp. **22** (1968), 742–762.

[CHS99] A. Cura Hochabaum and C. Schumann, *Free surface viscous flow around ship models*, Proc. of the 7th International Conference on Numerical Ship Hydrodynamics (Nantes), 1999.

[CL83] M. Crandall and P.-L. Lions, *Viscosity solutions of Hamilton–Jacobi equations*, Trans. Amer. Math. Soc. **277** (1983), 1–42.

[Con04] B. Conserva, *Simulazione numerica di flussi a superficie libera con applicazioni in idrodinamica navale*, Master thesis, Politecnico di Milano, 2004, In Italian.

[Cow01] G. W. Cowles, *A viscous multiblock flow solver for free-surface flows past complex geometries*, Ph.D. thesis, Princeton University, 2001.

[CPS03] G. W. Cowles, N. Parolini, and M. L. Sawley, *Numerical simulation using RANS-based tools for America's Cup design*, Proceedings of the 16th Chesapeake Sailing Yacht Symposium (USA Annapolis, ed.), 2003.

[CS02] R. Codina and O. Soto, *A numerical model to track two-fluid interfaces based on a stabilized finite element method and the level set technique*, Int. J. Num. Meth. Fluids **40** (2002), 293–301.

[Daw77a] C. W. Dawson, *Method for solving ship wave problems*, Proceedings of the 2nd International Conference on Numerical Ship Hydrodynamics, 1977, pp. 305–318.

[Daw77b] ———, *Numerical solutions of transient three-dimensional ship-wave problems*, Proceedings of the 2nd International Conference on Numerical Ship Hydrodynamics, 1977.

[DG04] L. Del Grosso, *Modellazione matematica e numerica della dinamica di una canoa da regata*, Master thesis, Politecnico di Milano, 2004, In Italian.

[DRRF02] F. Jr. DeBord, J. Reichel, B. Rosen, and C. Fassardi, *Design optimization for the International America's Cup Class*, Boston, 2002.

[dSMN+04] F. S. de Sousa, N. Mangiavacchi, L. G. Nonato, A. Castelo, M. F. Tomé, V. G. Ferreira, J. A. Cuminato, and S. McKee, *A front-tracking/front-capturing method for the simulation of 3D multi-fluid flows with free surfaces*, J. Comp. Phys. **198** (2004), no. 2, 469–499.

[DT80] A. Dervieux and F. Thomasset, *Approximation Methods for Navier–Stokes Problems*, Lecture Notes in Mathematics, vol. 771, ch. A finite element method for the simulation of Rayleigh-Taylor instability, pp. 145–158, Springer-Verlag, Berlin, 1980.

[EG04] A. Ern and J.-L. Guermond, *Theory and practice of finite elements*, Applied Mathematical Sciences, vol. 159, Springer-Verlag, 2004.

[ET98] A. Esmaeli and G Tryggvason, *Direct numerical simulations of bubbly flows. Part 1:Low Reynolds number arrays*, J. Fluid Mech. **377** (1998), 331–345.

[FF02] M. Falcone and R. Ferretti, *Semi-Lagrangian schemes for Hamilton-Jacobi equations, discrete representation formulae and Godunov methods*, J. Comput. Phys. **175** (2002), no. 2, 559–575.

[Flu03] *Fluent User's Manual*, Fluent Incorporated, Lebanon, NH, USA, 2003.

[FM04] L. Formaggia and E. Miglio, *Modello dinamico di una canoa*, Tech. report, MOX, Politecnico di Milano, Italia, 2004, In Italian.

[FMJ93] J. R. Farmer, L. Martinelli, and A. Jameson, *A fast multigrid method for solving incompressible hydrodynamic problems with free surfaces*, AIAA Journal **32** (1993), no. 6, 1175–1182.

[FVB99] G. Fekken, A. E. P. Veldman, and B. Buchner, *Simulation of the green water loading using the Navier–Stokes equations*, Proc. of the 7th International Conference on Numerical Ship Hydrodynamics (Nantes), 1999.

[GQ98] J. L. Guermond and L. Quartapelle, *On the approximation of the unsteady Navier–Stokes equations by finite element projection methods*, Numer. Math. **80** (1998), 207–238.

[GR86] V. Girault and P.A. Raviart, *Finite element methods for Navier-Stokes equations*, Springer Series in Comput. Math., vol. 5, Springer, Berlin Heidelberg New York, 1986.

[GRR04] S. Gross, V. Reichelt, and A. Reusken, *A finite element based level set method for two-phase incompressible flows*, 2004, IGPM Report Nr. 243.

[Gue99a] J.-L. Guermond, *Stabilization of Galerkin approximation of transport equations by subgrid modeling*, Math. Model. Numer. Anal. **33** (1999), 1293–1316.

[Gue99b] ———, *Un résultat de convergence à l'ordre deux en temps pour l'approximation des équations de Navier–Stokes par une technique de projection*, Modél. Math. Anal. Numér. (M^2AN) **33** (1999), 169–189.

[HB76] J. G. Hnat and J. D. Buckmaster, *Spherical cap bubbles and skirt formation*, Phys. Fluids **19** (1976), 182–194.

[HB82] T.J.R. Hughes and A. Brooks, *Finite elements in fluids*, vol. 4, ch. A theoretical framework for Petrov–Galerkin methods with discontinuous weighting functions: Application to the streamline upwind procedure, pp. 47–66, J. Wiley & Sons, Chichester, 1982.

[HD86] B. R. Hutchinson and Raithby G. D., *A multigrid method based on the additive correction strategy*, Num. Heat Trans. **9** (1986), 511–537.

[HEML00] T. J. R. Hughes, G. Engel, L. Mazzei, and M. G. Larson, *The continuous Galerkin method is locally conservative*, J. Comp. Phys. **163** (2000), 467–488.

[Hin92] T. Hino, *Computation of viscous flows with free surface around an advancing ship*, Proc. of the 2nd Osaka International Colloquium on Viscous Fluid Dynamics in Ship and Ocean Technology (Osaka), 1992.

[HN81] C. W. Hirt and B. D. Nichols, *Volume of Fluid (VOF) method for the dynamics of free boundaries*, J. Comp. Phys. **39** (1981), 201–225.

[HR88] J. G. Heywood and R. Rannacher, *Finite element approximation of the nonstationary Navier-Stokes problem, part iii. smoothing property and higher order error estimates for spatial discretization*, SIAM J. Numer. Anal. **25** (1988), no. 3, 489–512.

[HS67] J. L. Hess and A. M. O Smith, *Calculation of potential flow about arbitrary bodies*, vol. 8, pp. 1–138, 1967.

[HW65] F. H. Harlow and J. E. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluid with a free interface*, Physics of Fluids **8** (1965), 2182–2189.

[HW96] R. H. W. Hoppe and B. Wohlmuth, *Element-oriented and edge-oriented local error estimators for nonconforming finite element methods*, RAIRO Modél. Math. Anal. Numér. **30** (1996), no. 2, 237–263.

[Hym84] J. M. Hyman, *Numerical methods for tracking interfaces*, Physica **12D** (1984), 396–407.

[IC03] A Iafrati and E. F. Campana, *A domain decomposition approach to compute wave breaking (wave breaking flows)*, Int. J. Numer. Meth. Fluids **41** (2003), 419–445.

[IDMC01] A. Iafrati, A. Di Mascio, and E. F. Campana, *A level-set technique applied to unsteady free surface flows*, Int. J. Numer. Meth. Fluids **35** (2001), 281–297.

[IOnDP04] S. R. Idelsohn, E. Oñate, and F. Del Pin, *The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves*, Int. J. Num. Meth. Eng. **61** (2004), no. 7, 964–989.

[JK01] P. Jones and R. Korpus, *America's Cup class yacht design using viscous flow CFD*, Proceedings of the 16th Chesapeake Sailing Yacht Symposium (Annapolis,USA), 2001.

[Joh87] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, 1987.

[Kel86] Lord Kelvin, *On stationary waves in flowing water*, Phil. Mag. **22-23** (1886).

[KMI+83] H. Kajitani, H. Miyata, M. Ikehata, H. Tanaka, H. Adachi, M. Namimatzu, and S. Ogiwara, *Summary of the cooperative experiment on Wigley parabolic model in Japan*, Proc. of the 2nd DTNSRDC Workshop on Ship Wave Resistance Computations (Bethesda, USA), 1983, pp. 5–35.

[KP03] O. Karakashian and F. Pascal, *A-posteriori error estimates for a discontinuous Galerkin approximation of second order elliptic problems*, SIAM Jour. Num. Anal **41** (2003), no. 6, 2374–2399.

[KTP00] M. Kremenetsky, T. Tysinger, and S. Posey, *Considerations for parallel CFD enhancements on SGI ccNUMA and cluster architectures*, Parallel CFD Conference (Norway Trondheim, ed.), 2000.

[Lam93] H. Lamb, *Hydrodynamics*, 6th ed., Cambridge University Press, New York, NY, 1993.

[Lar90] L. Larsson, *Scientific methods in yacht design*, Ann. Rew. Fluid Mech. **22** (1990), 349–385.

[Laz97] L. Lazauskas, *A performance prediction model for rowing races*, Tech. Report L9702, Dept. of Appl. Math., University of Adelaide, Australia, 1997, Technical Report.

[LeV92] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Second edition, Birkhäuser Verlag , 1992.

[Lio97] P. L. Lions, *Mathematical topics in fluid mechanics: Incompressible models*, Oxford Lecture Series in Mathematics and Its Applications, vol. 3, Oxford, 1997.

[LL97] R. J. LeVeque and Z. Li, *Immersed interface methods for stokes flow with elastic boundaries or surface tension*, SIAM J. Sci. Comput. **18** (1997), no. 3, 709–735.

[LML+04] R.B. Langtry, F.R. Menter, S.R. Likki, Y.B. Suzen, P.G. Huang, and S. Vlker, *A correlation based transition model using local variables. Part 2: Test cases and industrial applications*, Proceedings of ASME Turbo. Expo. (Vienna, Austria), 2004.

[LR74] P. Lesaint and P. A. Raviart, *Mathematical aspects of finite elements in partial differential problems*, ch. On a Finite Element Method for Neutron Transport Equation, pp. 89–119, Academic Press, 1974.

[LRR75] D. B. Launder, G.J. Reece, and W. Rodi, *Progress in development of a Reynolds stress turbulence closure*, J. Fluid Mech. **68** (1975), 537–566.

[LS72] D. B. Launder and D. B. Spalding, *Lectures in mathematical models of turbulence*, Academic Press, London, 1972.

[LYOnI99] R. Lohner, C. Yang, E. Oñate, and S.R. Idelsohn, *An unstructured grid-based, parallel free surface solver*, Appl. Num. Math. **31** (1999), no. 3, 271–293.

[Mic98] J. H. Michell, *The wave resistance of a ship*, Phil. Mag. **45** (1898), 106–123.

[Mil87] A. Millward, *A study of the forces exerted by an oarsman and the effect on boat speed*, J. Sports Sciences **5** (1987), 93–103.

[Mil93] J. H. Milgram, *Naval architecture technology used in winning the 1992 America's Cup match*, Trans. Soc. Naval Arch. Mar. Eng. **101** (1993), 399–436.

[Mil98] ———, *Fluid mechanics for sailing vessel design*, Ann. Rew. Fluid Mech. **30** (1998), 613–653.

[Miy86] H. Miyata, *Finite-difference simulation of breaking waves*, J. Comp. Phys. **65** (1986), 179–214.

[MLL+04] F.R. Menter, R.B. Langtry, S.R. Likki, Y.B. Suzen, P.G. Huang, and S. Vlker, *A correlation based transition model using local variables. Part 1: Model formulation*, Proceedings of ASME Turbo. Expo. (Vienna, Austria), 2004.

[MLS01] B. Metcalf, J. Longo, and F. Stern, *Experimental investigation of wave-induced separation around a surface-piercing NACA0024 hydrofoil*, Proc. of the 26th ATTC Conference (Webb Institute, Glen Cove, N.Y.), 2001.

[MM52] J. C. Martin and W. J. Moyce, *An experimental study of the collapse of liquid columns on a rigid horizontal plane*, Philos. Trans. R. Soc. **244** (1952), 312–324.

[Mol04] A. Mola, *Computational fluid dynamics for flows around slender bodies with appendages*, Master thesis, Politecnico di Milano, 2004.

[Mor96] K. W. Morton, *Numerical solution of convection-diffusion problems*, Chapman & Hall, London, 1996.

[MP94] P. Mohammadi and O. Pironneau, *Analysis of the k-epsilon model*, Masson, Paris, 1994.

[MP97] S. Muzaferija and M. Peric, *Computation of free-surface flows using finite volume method and moving grids*, Numer. Heat Trans., Part B **32** (1997), 369–384.

[MP99] ———, *Nonlinear water wave interaction*, ch. Computation of free surface flows using interface-tracking and interface-capturing methods, pp. 59–110, WIT Press, Southampton, 1999.

[MPR99] V. Maronnier, M. Picasso, and J. Rappaz, *Numerical simulation of free surface flows*, J. Comp. Phys. **155** (1999), no. 2, 439–455.

[MPR03] ———, *Numerical simulation of three-dimensional free surface flows*, Int. J. Numer. Meth. Fluid **42** (2003), no. 7, 697–716.

[NH73] B. D. Nichols and C. W. Hirt, *Improved free surface boundary conditions for numerical incompressible flow calculation*, J. Comp. Phys. **8** (1973), 434–448.

[Nie03] K. B. Nielsen, *Numerical prediction of green water loads on ships*, Ph.D. thesis, Technical University of Denmark, 2003.

[NM96] T. Nakayama and M. Mori, *An Eulerian finite element method for time-dependent free surface problems in hydrodynamics*, Int. J. Num. Meth. Fluids **22** (1996), 175–194.

[NW76] W. F. Noh and P. Woodward, *SLIC (Simple Line Interface Calculation)*, Proc. of the 5th International Conference on Fluid Dynamics (Berlin) (A.I. van de Vooren and P.J. Zandbergen, eds.), Lecture Notes in Physics, vol. 59, Springer, 1976.

[OF02] S. Osher and R. Fedkiw, *The level set method and dynamic implicit surfaces*, Springer-Verlag, New York, 2002.

[OS88] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: Algorithm based on Hamilton–Jacobi formulations*, J. Comp. Phys. **79** (1988), 12–49.

[OS91] S. Osher and C.-W. Shu, *High-order essentially non-oscillatory schemes for Hamilton–Jacobi equations*, SIAM J. Numer. Anal. **28** (1991), 902–921.

[Osw91] P. Oswald, *On a bpx-preconditionner for p1 elements*, 1991, Preprint, FSU Jena.

[Pan04] S. Pandini, *Simulazione numerica della dinamica di una imbarcazione da canottaggio sportivo*, Master thesis, Politecnico di Milano, 2004, In Italian.

[Pat80] S. V. Patankar, *Numerical heat transfer and fluid flow*, Hemisphere, Washington, D.C., 1980.

[Per93] J. B. Perot, *An analysis of the fractional step method*, J. Comput. Phys. **108** (1993), no. 1, 51–58.

[PMO+99] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, *A pde-based fast local level set method*, J. Comp. Phys. **155** (1999), no. 2, 410–438.

[PP04] J. E. Pilliot and E. G. Puckett, *Second-order accurate volume-of-fluid algorithms for tracking material interfaces*, J. Copm.Phys. **199** (2004), no. 2, 465–502.

[PQ04] N. Parolini and A. Quarteroni, *Mathematical models and numerical simulations for the America's Cup*, Comp. Meth. Appl. Mech. Eng. (2004), available online.

[PZ99] S. Popinet and S. Zaleski, *A front-tracking algorithm for accurate representation of surface tension*, Int. J. Numer. Meth. Fluids **30** (1999), 775–793.

[QSV99] A. Quarteroni, F. Saleri, and A. Veneziani, *Analysis of the Yosida method for the incompressible Navier-Stokes equations*, J. Math. Pures Appl. (9) **78** (1999), no. 5, 473–503.

[QSV00] ———, *Factorization methods for the numerical approximation of Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg. **188** (2000), no. 1-3, 505–526.

[Qua93] L. Quartapelle, *Numerical solution of the incompressible Navier-Stokes equations*, International Series of Numerical Mathematics, vol. 113, Birkhäuser Verlag, Basel, 1993.

[QV94] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer Series in Computational Mathematics, vol. 23, Springer-Verlag, Berlin, 1994.

[RL83] C. M. Rhie and Chow W. L., *Numerical study of the turbulent flow past an airfoil with trailing edge separation*, AIAA Journal **21** (1983), no. 11, 1525–1532.

[RLDS93] B. S. Rosen, J. P. Laiosa, W. H. Davis, and D. Stavetski, *Splash free-surface code methodology for hydrodynamic design and analysis of IACC yachts*, Proc. of 11th Chesapeake Sailing Yacht Symposium (Annapolis), 1993.

[RT76] J. D. Ramshaw and J. A. Trapp, *A numerical technique for low-speed homogeneous two-phase flow with sharp interface*, J. Comp. Phys. **21** (1976), 438–453.

[Rud97] M. Rudman, *Volume-tracking methods for interfacial flow calculations*, Int. J. Num. Meth. Fluids **24** (1997), 671–691.

[SA92] P. Spalart and S. Allmaras, *A one-equation turbulence model for aerodynamic flows*, 30th Aerospace Sciences Meeting, Reno, 1992, AIAA paper 92-0439.

[Sch68] H. Schlichting, *Boundary Layer Theory*, 5th ed., McGraw-Hill, New York, 1968.

[SD00] M. Sussman and D. Dommermuth, *The numerical simulation of ship waves using cartesian grid methods*, Proc. of the 23rd Symposium on Naval Hydrodynamics (Val de Rueil, France), 2000.

[Set99] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 1999.

[SF99] M. Sussman and E. Fatemi, *An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow*, SIAM J. Sci. Comput. **20** (1999), no. 4, 1165–1191.

[Sil94] D. Silvester, *Optimal low order finite element methods for incompressible flow*, Comput. Methods Appl. Mech. Engrg. **111** (1994), 357–368.

[SLS+95] T.-H. Shih, W. W. Liou, A. Shabbir, Z. Yang, and J. Zhu, *A new eddy-viscosity model for high Reynolds number turbulent flows - model development and validation*, Computers and Fluids **24** (1995), no. 3, 227–238.

[Smo01] A. Smolianski, *Numerical modeling of two-fluid interfacial flows*, Ph.D. thesis, University of Jyväskylä, Finland, 2001.

[SN93] C. A. Scragg and B. D. Nelson, *The design of an eight-oared rowing shell*, Marine Tech. **30** (1993), no. 2, 84–99.

[Spe01] T. S. Speer, *The BASILISCUS Project - Return of the cruising hydrofoil sailboat*, Proceedings of the 15th Chesapeake Sailing Yacht Symposium (Annapolis, USA), 2001.

[SS97] M. Sussman and P. Smereka, *Axisymmetric free boundary problems*, J. Fluid Mech. **341** (1997), 269–294.

[SSO94] M. Sussman, P. Smereka, and S. Osher, *A level set approach for computing solutions to incompressible two-phase flow*, J. Comp. Phys. **114** (1994), 146–159.

[SV] F. Saleri and A. Veneziani, *Pressure-correction algebraic splitting methods for the incompressible Navier–Stokes equations*, Quaderno MOX, n. 26, Politecnico di Milano, 2003.

[TBE+01] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y. J. Jan, *A front-tracking method for the computations of multiphase flows*, J. Comp. Phys. **169** (2001), 708–759.

[THD01] S. R. Turnock, N. J. Holroyd, and J. C. Date, *Appendage design for the America's Cup using CFD*, Proceedings of ECCOMAS CFD 2001 (Swansea), 2001.

[Tho84] V. Thomée, *Galerkin finite element methods for parabolic problems*, Lecture Notes in Mathematics, vol. 1054, Springer, New York, 1984.

[TLW95] T. G. Thomas, D. C. Leslie, and J. J. R. Williams, *Free surface simulations using a conservative 3D code*, J. Comp. Phys. **116** (1995), 52–68.

[TM94] M. Tomé and S. McKee, *GENSMAC: A computational Marker-And-Cell method for free surface flows in general domains*, J. Comp. Phys. **110** (1994), 171–186.

[Tor00] A.-K. Tornberg, *Interface tracking methods with applications to multiphase flows*, Ph.D. thesis, Royal Institute of Technology, Stockholm, Sweden, 2000.

[TT01] M. Tabata and D. Tagami, *A finite element analysis of a linearized problem of the Navier–Stokes equations with surface tension*, SIAM J. Numer. Anal. **38** (2001), no. 1, 40–57.

[TU90] G. Tryggvason and S. O. Unverdi, *Computations of three-dimensional Rayleigh–Taylor instability*, Phys. Fluids **2** (1990), 656–671.

[UT92] S. O. Unverdi and G. Tryggvason, *A front-tracking method for viscous, incompressible, multi-fluid flows*, J. Comp. Phys. **100** (1992), 25–37.

[vdPSV03] S. P. van der Pijl, A. Segal, and C. Vuik, *A mass-conserving level-set (MCLS) method for modeling of multi-phase flows*, 2003, Report 03-03.

[VL99] M. Vogt and L. Larsson, *Two level set methods for predicting viscous free surface flows*, Proceedings of the 7th International Conference on Numerical Ship Hydrodynamics (Nantes), 1999.

[Wil93] D. C. Wilcox, *Turbulence modeling for CFD*, DCW Industries, 1993.

[Wol69] M. Wolfstein, *The velocity and temperature distribution of one-dimensional flow with turbulence augmentation and pressure gradient*, Int. J. Heat Mass Trans. **12** (1969), 301–318.

[ZCMO96] H. Zhao, T. Chan, B. Merriman, and S. Osher, *A variational level set approach to multiphase motion*, J. Comp. Phys. **127** (1996), 179–195.

[Zho97] G. Zhou, *How accurate is the streamline diffusion finite element method?*, Math. Comp. **66** (1997), 31–44.

Curriculum Vitæ

Nicola Parolini est né le 9 julliet 1975 à Vimercate (MI), Italie. Il a effectué son école secondaire au Lycée M. Curie à Tradate (VA) où il a obtenu en 1994 son Baccalauréat Scientifique. Ensuite, il a poursuivi ses études à l'Ecole Polytechnique de Milan pour obtenir en 2000 un diplôme d'ingénieur aérospatial. Depuis juillet 2001 il a été engagé comme assitant dans la Chaire de Modélisation et Calcul Scientifique du Professeur Alfio Quarteroni. Il a participé à plusieurs conférences internationales et il est co-auteur des publications suivantes :

- E. BURMAN AND N. PAROLINI, Local Edge Stabilization for the Transport Equation. In preparation, 2004
- E. BURMAN AND N. PAROLINI, A New Reinitialization Procedure for the Finite Element Approximation of the Level Set Equation. In preparation, 2004
- N. PAROLINI AND E. BURMAN, A Finite Element Level Set Method for Viscous Free-Surface Flows. To appear in *Proceedings of SIMAI 2004*, Venezia, Italy.
- N. PAROLINI AND A. QUARTERONI, Mathematical Models and Numerical Simulations for the America's Cup, *Comp. Meth. Appl. Mech. Eng.*. In press, available on-line.
- N. PAROLINI AND A. QUARTERONI, Simulazione Numerica per la Coppa America di Vela, *Bollettino UMI, La Matematica nella Società e nella Cultura*, Serie VIII, Vol. 7-A, 1–15 (2004).
- N. PAROLINI AND A. QUARTERONI, Numerical Simulation for Yacht Design, in *Proceedings of the 6th Conference on Informatics and Mathematics, HERCMA 2003* (E.A. Lipitakis, ed.), Vol. 1, pp. 38-44, Athens, 2004.
- F. AUTERI, N. PAROLINI AND L. QUARTAPELLE, Essential imposition of the Neumann condition in Galerkin–Legendre elliptic solvers, *J. Comput. Phys.*, **185**, 427–444 (2003).
- G. W. COWLES, N. PAROLINI AND M. L. SAWLEY, Numerical Simulation using RANS-based Tools for America's Cup Design, in *Proceedings of the 16th Chesapeake Sailing Yacht Symposium, Annapolis*, J. Zseleczky (ed), Omnipress, Madison (USA) (2003).
- A. QUARTERONI, M. SALA, M. L. SAWLEY, N. PAROLINI AND G. W. COWLES, Mathematical Modelling and Visualisation of Complex Three-Dimensional Flows, in *Visualisation and Mathematics III*, Springer-Verlag series "Mathematics and Visualization", H.-C. Hege and K. Polthier (eds), 361-377, Heidelberg 2003.
- F. AUTERI, N. PAROLINI AND L. QUARTAPELLE, Numerical investigation on the stability of the singular driven cavity flow, *J. Comput. Phys.*, **183**, 1-25 (2002).

- F. AUTERI AND N. PAROLINI, Numerical investigation of the first instabilities in the differentially heated 8:1 cavity, *Int. J. Numer. Meth. Fluids* **40**, 1121–1132 (2002).
- F. AUTERI AND N. PAROLINI, A mixed-basis spectral projection method, *J. Comput. Phys.* **175**, 1–23 (2002).
- F. AUTERI AND N. PAROLINI, Simulation of the differentially-heated 8:1 rectangular cavity by a Galerkin–Legendre spectral projection method, *Proceedings of the First M.I.T. Conference on Computational Fluid and Solid Mechanics*, Cambridge, USA (2001).
- F. AUTERI, J.-L. GUERMOND AND N. PAROLINI, Role of the LBB condition in weak spectral projection methods, *J. Comput. Phys.* **174**, 405–420 (2001).